

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS



DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO DE SISTEMAS Y COMPUTACIÓN

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
CONTABLE BÁSICO PARA USO LIBRE EN
PLATAFORMA WEB”**

AUTOR:

EMANUEL RICARDO CARRASCO VÁSQUEZ

DIRECTOR: JAVIER W. CÓNDOR

QUITO, 2011 Agosto

INDICE

DEDICATORIA	viii
AGRADECIMIENTO	viii
INTRODUCCIÓN	viii
CAPÍTULO 1: MARCO TEÓRICO	1
1. EL INTERNET.....	1
1.1 ORIGENES DEL INTERNET, RESUMEN HISTÓRICO	1
1.2 WORLD WIDE WEB	2
1.3 INTERNET, SU EVOLUCIÓN Y ACTUALIDAD	3
1.4 EL FUTURO: EL INTERNET 2.....	4
2. HTML	5
2.1 Elementos	6
2.2 Cronología del HTML	6
2.3 FUNCIONAMIENTO DE LA WEB	8
3. SOFTWARE LIBRE	10
3.1 Libertades del software libre	10
3.2 TIPOS DE LICENCIAS.....	11
3.2.1 Licencias GPL.....	11
3.2.2 Licencias AGPL	11
3.2.3 Licencias estilo BSD	11
3.2.4 Licencias estilo MPL y derivadas.....	12
3.2.5 Copyleft.....	12
3.3 Comparación con el software de código abierto	12
3.4 Ventajas del software libre	13
3.5 Impacto del software libre	13
4. PHP.....	14
4.1 Ventajas	16
4.2 Las cuatro grandes características de PHP	17
4.3 Inconvenientes	18
5. AJAX	18
5.1 Ventajas	21

5.2 Problemas e Inconvenientes	21
6. MySQL	22
6.1 Características distintivas.....	22
6.2 Tipos de compilación del servidor	24
7. CLOUD COMPUTING: LA NUEVA ERA DEL DESARROLLO	24
7.1 INTRODUCCIÓN	25
7.2 CARACTERÍSTICAS DEL CLOUD COMPUTING	26
7.3 VENTAJAS.....	27
7.4 DESVENTAJAS	28
7.5 CAPAS	29
7.5.1 Software como un Servicio (SaaS).....	29
7.5.2 Plataforma como un Servicio (PaaS).....	29
7.5.3 Infraestructura como un Servicio (IaaS).	29
7.6 TIPOS DE NUBES	30
7.7 EJEMPLOS DE CLOUD COMPUTING.....	30
8. ARQUITECTURA ORIENTADA A SERVICIOS (SOA).....	31
8.1 HISTORIA	32
8.2 BENEFICIOS.....	33
8.3 DESVENTAJA.....	33
9. METODOLOGÍAS AGILES.....	34
9.1 PROGRAMACIÓN EXTREMA (EXTREME PROGRAMMING, XP).....	35
9.2 VALORES	36
9.3 ROLES DEFINIDOS EN LA PROGRAMACIÓN EXTREMA.....	38
9.3.1 Programador.....	38
9.3.2 Cliente.....	38
9.3.3 Encargado de pruebas (Tester)	38
9.3.4 Encargado de seguimiento (Tracker)	38
9.3.5 Entrenador (Coach)	38
9.3.6 Consultor	39
9.3.7 Gestor (Big boss).....	39
9.4 CARACTERÍSTICAS DEL PROCESO DE XP	39
9.5 CICLO DE VIDA IDEAL DE XP:.....	40
9.5.1 Fase I: Exploración.....	40

9.5.2 Fase II: Planificación de la Entrega	41
9.5.3 Fase III: Iteraciones	41
9.5.4 Fase IV: Producción	42
9.5.5 Fase V: Mantenimiento.....	42
9.5.6 Fase VI: Muerte del Proyecto	42
10. Contabilidad	43
10.1 CONTABILIDAD FINANCIERA.....	44
10.2 CONTABILIDAD DIRECTIVA.....	44
10.3 CONTABILIDAD BÁSICA.....	44
CAPÍTULO 2: ANÁLISIS Y DESARROLLO	45
1. IDENTIFICACIÓN DE REQUERIMIENTOS	45
1.1 REQUERIMIENTOS FUNCIONALES	45
F0: Ingreso al Sistema.....	45
F1: Administración para el manejo de Usuarios	45
F2: Administración para el manejo de Cuentas Contables.....	45
F3: Administración para el manejo de Periodos.....	45
F4: Administración para el manejo de Asientos Contables	45
F5: Procesos prioritarios del Sistema Contable.....	46
F6: Reportes	46
F7: Salir del Sistema.....	46
1.2 REQUERIMIENTOS NO FUNCIONALES.....	46
2. HERRAMIENTAS DEL SISTEMA.....	46
3. UML y Diagramas	47
3.1 DIAGRAMAS DE CASOS DE USO	48
3.1.1 DIAGRAMA GENERAL (NIVEL 0).....	48
3.1.2 DIAGRAMA A DETALLE	49
F1: Administración para el manejo de Usuarios	49
F1.1: Ingreso de Usuarios	49
F1.2: Actualización de Usuarios	50
F1.3: Eliminación de Usuarios	50
3.2 DIAGRAMA DE CLASES	51
3.2.1 DIAGRAMA GENERAL	51
4. MODELO ENTIDAD – RELACIÓN	52

4.1 MODELO CONCEPTUAL	52
4.2 MODELO FÍSICO.....	53
5. PROTOTIPO DE INTERFAZ	54
5.1 LOGIN DE USUARIOS	54
5.2 PROTOTIPO DE INTERFAZ.....	54
5.3 PROTOTIPO DE ERROR	55
5.4 PROTOTIPO DE ERROR DE CAMPOS.....	55
5.5 PROTOTIPO DE REPORTE.....	56
CAPÍTULO 3: IMPLEMENTACION.....	57
1. DESARROLLO Y OBSERVACIONES V1.0	57
1.1 DESARROLLO V1.0	57
1.2 OBSERVACIONES DEL CLIENTE DE PRUEBA V1.0.....	57
1.3 DESARROLLO V1.5	60
1.4 OBSERVACIONES DEL CLIENTE DE PRUEBA V1.5.....	60
1.5 DESARROLLO V2.0	60
CAPÍTULO 4: PRUEBAS Y SERVICIO	61
1. PRUEBAS.....	61
1.1 PRUEBAS UNITARIAS	61
1.2 PRUEBAS CONJUNTAS	61
1.3 PRUEBAS CONCURRENTES.....	62
2. SERVICIO.....	62
2.1 SERVICIO EN VIVO.....	62
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.....	63
1. CONCLUSIONES	63
1.1 CONCLUSIONES DEL CAPÍTULO I.....	63
1.2 CONCLUSIONES DEL CAPÍTULO II.....	64
1.3 CONCLUSIONES DEL SISTEMA DESARROLLADO	64
1.4 CONCLUSIONES QUE VALIDEN LA HIPÓTESIS.....	65
1.5 RECOMENDACIONES PARA EL USO DEL SISTEMA	65
1.6 RECOMENDACIONES PARA EL LEVANTAMIENTO DE REQUERIMIENTOS.....	66
1.7 RECOMENDACIONES TÉCNICAS	66
1.8 RECOMENDACIONES GENERALES	67
BIBLIOGRAFÍA.....	68

OTRAS REFERENCIAS	69
ANEXOS	70
1. DIAGRAMAS DE CASOS DE USO A DETALLE – OTRAS ADMINISTRACIONES Y PROCESOS DEL SISTEMA	70
F2: Administración para el manejo de Cuentas Contables.....	70
F2.1: Ingreso del de Cuentas Contables	70
F2.2: Actualización de Cuentas Contables	71
F2.3: Eliminación de Cuentas Contables.....	71
F3: Administración para el manejo de Periodos	71
F3.1: Actualización de Periodos	72
F4: Administración para el manejo de Asientos Contables	72
F4.1: Ingreso de Asientos Contables.....	73
F4.2: Actualización de Asientos Contables	73
F4.3: Eliminación de Asientos Contables	73
F4.4: Cuadre de Valores.....	74
F5: Procesos prioritarios del Sistema Contable.....	74
F5.1: Mayorización	75
F5.2: Cierre de Mes	75
F5.3: Cierre Anual.....	75
F6: Reportes	76
F6.1: Balance Mensual	76
F6.2: Balance Anual	76

INDICE DE FIGURAS

FIG 1-01 World Wide Web.....	3
FIG 1-02 Estructura general de una etiqueta HTML.....	6
FIG 1-03 Diagrama de Funcionamiento de la Web	9
FIG 1-04 Mapa conceptual del software libre y de código abierto.....	12
FIG 1-05 Esquema del funcionamiento de las páginas PHP.....	15
FIG 1-06 Características de PHP	18
FIG 1-07 Modelo AJAX de aplicaciones Web	19
FIG 1-08 Modelo AJAX de aplicaciones Web II.....	20
FIG 1-09 Evolución y versionamiento de MySQL.....	23
FIG 1-10 Esquema principal de Cloud Computing	24
FIG 1-11 Cloud Computing orientado al servicio	25
FIG 1-12 Evolución de Cloud Computing	26
FIG 1-13 Infraestructura como un servicio	29
FIG 1-14 Estructura SOA.....	31
FIG 1-15 Proceso de Desarrollo	39
FIG 1-16 Ciclo de vida en V	40
FIG 1-17 Las prácticas se refuerzan entre sí	43
FIG 2-01 Mapa Mental UML	47
FIG 2-02 Diagrama de Casos de Uso General	48
FIG 2-03 Diagrama de Casos de Uso – Administración de Usuarios	49
FIG 2-04 Diagrama de Casos de Uso – Ingreso de Usuarios	49
FIG 2-05 Diagrama de Casos de Uso – Actualización de Usuarios	50
FIG 2-06 Diagrama de Casos de Uso – Eliminación de Usuarios	50
FIG 2-07 Diagrama de Clases General	51

FIG 2-08 Modelo Conceptual	52
FIG 2-09 Modelo Físico	53
FIG 2-10 Ingreso de Usuarios	54
FIG 2-11 Interfaz Estándar	54
FIG 2-12 Prototipo de visualización de errores	55
FIG 2-13 Prototipo de visualización de errores por campo	55
FIG 2-14 Prototipo de visualización de reportes	56

INDICE DE TABLAS

Tabla 1-01 Libertades del Software Libre	10
Tabla 1-02 Comparación Metodologías Ágiles y Metodologías Tradicionales	34
Tabla 1-03 Ranking de Agilidad entre las diferentes metodologías	35

DEDICATORIA

Esta disertación está dedicada a todos aquellos familiares que hicieron posible que pueda culminar mis estudios universitarios, en especial a mi madre María Vásconez, que con sus esfuerzos y sacrificios me formó como persona y me permitió valorar que la vida por más que es dura, uno siempre puede seguir adelante a pesar de la adversidad.

AGRADECIMIENTO

Un especial agradecimiento a mi hermano y guía José Miguel, él quien más que ser un hermano es un amigo, me enseñó prácticamente todo lo que se refiere a sistemas computacionales. Gracias a él estudié la carrera de Ingeniería en Sistemas y por él es que puedo conocer sobre las nuevas tecnologías del mercado. A pesar de la distancia y el reconocimiento internacional que se ha ganado por su esfuerzo y sacrificio, siempre supo apoyarme y estar en los momentos más difíciles de mi vida.

Agradezco a mi familia en general por brindarme su apoyo en cada momento de mi vida y a aquellos que han puesto un grano de arena para cumplir con los objetivos de mi vida.

Es importante mencionar y agradecer también a todos los ingenieros y profesores de la carrera, quienes en algún momento ayudaron a mi formación académica y a mis amigos y conocidos, que han pasado por mi vida en estos años de estudio.

Para concluir, quiero nombrar a mi director de tesis y amigo Ing. Javier Cóndor, con quien hemos trabajado y compartido conocimientos para lograr culminar esta disertación de grado.

INTRODUCCIÓN

Los cambios tecnológicos ocurrentes en la actualidad, nos han llevado a evolucionar en la forma de desarrollar software, de forma que podamos utilizar al internet como un arma crítica e imprescindible en las nuevas aplicaciones del mercado. Es por este motivo que la presente disertación de grado se ha enfocado en un nuevo concepto de desarrollo, el Software como un servicio (SaaS). Este modelo de distribución en donde el software y los datos se encuentran en un servidor de la empresa desarrolladora y se accede a esta información a través de internet, está brindando innumerables ventajas a las empresas y a las personas.

CAPÍTULO 1: MARCO TEÓRICO

1.1 EL INTERNET

El Internet es una red de comunicación descentralizada, que para permitir la conexión entre computadoras utiliza el protocolo de comunicación TCP/IP¹, esto garantiza que “las redes físicas heterogéneas que la componen, funcionen como una red lógica única, de alcance mundial”². En otras palabras, y de forma más clara, decimos que el Internet es una red de redes o un conjunto de computadoras conectadas entre sí, que comparten una gran cantidad de datos e información.

1.1.1 ORIGENES DEL INTERNET, RESUMEN HISTÓRICO

El Internet ha logrado una revolución sin precedentes en el mundo de la informática y de las comunicaciones. Algunos de los inventos del pasado como el telégrafo, el teléfono, la radio y el ordenador sentaron las bases para ésta integración de capacidades nunca antes vivida. El Internet es a la vez una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre las personas y sus ordenadores independientemente de su localización geográfica.

Cronología:

1961: Leonard Kleinrock publicó el primer documento sobre la teoría de conmutación de paquetes. Kleinrock convence a Lawrence Roberts de la factibilidad teórica de las comunicaciones vía paquetes³ en lugar de circuitos.

1965: Roberts conecta una computadora TX2 en Massachusetts con un Q-32 en California a través de una línea telefónica de baja velocidad. Al conectar la TX2 con la Q-32 se crea la primera red de área amplia jamás construida hasta ese entonces.

1969: Nace la primera red interconectada, cuando se crea el primer enlace entre las universidades de UCLA y Stanford por medio de la línea telefónica conmutada, gracias a los trabajos y estudios anteriores de varios científicos y organizaciones desde 1959 (ARPANET⁴).

¹ TCP/IP: Transmission Control Protocol / Protocolo de Internet.

² <http://es.wikipedia.org/wiki/Internet>

³ Paquetes: Unidades de transporte de información.

⁴ ARPANET: Advanced Research Projects Agency Network.

1972: Se realizó la primera demostración pública de ARPANET, una nueva red de comunicaciones financiada por la DARPA⁵ que funcionaba de forma distribuida sobre la red telefónica conmutada.

1973: La DARPA inicia un programa de investigación sobre posibles técnicas para interconectar redes de distintas clases. Para este fin, desarrollaron nuevos protocolos⁶ de comunicaciones que permitiesen este intercambio de información de forma transparente para las computadoras conectadas.

En este mismo año se comenzó a estudiar la factibilidad técnica de desarrollar redes interconectadas, a base de los protocolos TCP y el IP. Fue durante aquel proceso investigativo, que se creó la palabra Internet, para apodar a estas redes.

1983: ARPANET cambia el protocolo NCP por TCP/IP. Ese mismo año, se crea el IAB con el fin de estandarizar el protocolo TCP/IP y de proporcionar recursos de investigación a Internet.

1986: La NSF⁷ comienza el desarrollo de NSFNET que se convierte en la principal Red en árbol de Internet, complementada después con las redes NSINET y ESNET, todas ellas en Estados Unidos. Paralelamente, otras redes troncales en Europa, tanto públicas como comerciales, junto con las americanas formaban el esqueleto básico de Internet.

1989: Con la integración de los protocolos OSI en la arquitectura de Internet, se inicia la tendencia actual de permitir no sólo la interconexión de redes de estructuras dispares, sino también la de facilitar el uso de distintos protocolos de comunicaciones.

1.1.2 WORLD WIDE WEB

Uno de los factores principales “que ha influenciado significativamente en la popularidad del Internet es el World Wide Web”⁸ en inglés. La WWW es un conjunto de protocolos que ayuda, de forma sencilla, la consulta remota de archivos de hipertexto, debido a que permite

⁵ DARPA: Defense Advanced Research Projects Agency.

⁶ Protocolos: Conjunto de normas que regulan la comunicación entre los distintos componentes de una red.

⁷ NSF: National Science Foundation.

⁸ <http://www.civila.com/desenredada/que-es.html>

desplegar gráficos y usar el mouse para navegar (visitar) los distintos lugares que existen en la red utilizando el Internet como medio de transmisión.



FIG 1-01 World Wide Web [A]

Antes el acceso era complicado y aburrido: en las pantallas sólo se mostraban textos y se debía usar instrucciones complicadas o programas manejados con el teclado.

Ahora se puede ir de un lado a otro, tan sólo seleccionando con el mouse en la pantalla un texto o gráfico gracias a lo que se conoce como las facilidades de hipertexto e hipermedia.

Si queremos definir a la WWW de forma más técnica, decimos que es un sistema de información distribuido basado en hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, videos u otros contenidos multimedia, y navegar a través de ellas usando links.

1.1.3 INTERNET, SU EVOLUCIÓN Y ACTUALIDAD

Inicialmente Internet tenía un objetivo claro, navegar para encontrar algo muy concreto, es decir, búsquedas de información. Ahora quizás también, pero sin duda alguna, hoy es más probable perderse en la red, debido al inmenso abanico de posibilidades que brinda. Hoy en día, la sensación que produce Internet es, cúmulo de ideas distintas, de personas diferentes, de pensamientos distintos de tantas y tantas posibilidades que, en ocasiones, puede resultar excesivo, “el internet es una serie de blogs”⁹.

El hecho de que Internet haya aumentado tanto implica una mayor cantidad de relaciones virtuales entre personas. Conociendo este hecho y relacionándolo con la felicidad originada por las relaciones personales, es posible concluir que cuando una persona tenga una

⁹ Vía: <http://es.engadget.com>

necesidad de conocimiento popular o de conocimiento no escrito en libros, puede recurrir a una fuente más acorde a su necesidad.

Internet incluye aproximadamente 5.000 redes en todo el mundo y más de 100 protocolos distintos basados en TCP/IP, que se configura como el protocolo de la red. Los servicios disponibles en la red mundial de PC, han avanzado mucho gracias a las nuevas tecnologías de transmisión de alta velocidad. Con el Internet se ha logrado unir a las personas mediante videoconferencia, imágenes por satélite, se ha podido observar el mundo por cámaras web, hacer llamadas telefónicas gratuitas, o disfrutar de un juego multijugador en 3D, un buen libro PDF, o álbumes y películas para descargar.

El abaratamiento de los costes de conexión, las mejoras tecnológicas y la llegada masiva de los ordenadores personales a los hogares, está produciendo un crecimiento exponencial de los usuarios y haciendo que éste se convierta en un instrumento imprescindible para ocio y negocio.

Desde el punto de vista de la comunicación permite crear grupos, asociaciones y comunidades virtuales con un objetivo o interés en común que solo es posible en este medio ya que elimina las distancias geográficas. A esto es a lo que llamamos globalización.

Hoy en día, nadie se puede quedar fuera de esta red de redes. Esta está presente en casi todos los hogares del mundo, como en todas las empresas del globo, ya de manera obligatoria. Muchos estudios, trabajos y funciones empresariales, no se conciben sin el Internet. Así de simple. Incluso muchos gobiernos, se han puesto en campaña, para alfabetizar digitalmente a sus ciudadanos. Es que la tecnología y el Internet, ya forma parte de nuestras vidas.

1.1.4 EL FUTURO: EL INTERNET 2

Internet 2 es el futuro de la red de redes y está formado actualmente por un consorcio dirigido por 206 universidades que junto a la industria de comunicaciones y el gobierno están desarrollando nuevas técnicas de conexión que acelerarán la capacidad de transferencia entre servidores.

Sus objetivos están enfocados a la educación y la investigación académica. Además buscan aprovechar aplicaciones de audio y video que demandan más capacidad de transferencia de ancho de banda.

“El proyecto Internet 2 (I2) es un esfuerzo de colaboración para desarrollar tecnología y aplicaciones avanzadas en el Internet, considerados vitales para las misiones de investigación y educación de las instituciones de educación superiores.”¹⁰

Este proyecto comenzó en el año de 1996 y ha sido desarrollado por el consorcio con el fin de realizar software como la telemedicina, bibliotecas digitales y laboratorios virtuales, que requieren conexiones de gran velocidad en tiempo real.

El Internet 2 opera sobre una red desarrollada principalmente por las universidades estadounidenses, la cual está construida con fibra óptica y que permite una excelente fiabilidad.

Sin embargo, no hay que pensar que esta nueva red reemplazará a la Internet actual. La herencia de Internet 2, será la tecnología adoptada y desplegada por las redes de comunicación comerciales, tal como sucedió con la Internet de hoy que salió de las redes de investigación federales y académicas en la década de 1980.

1.2. HTML

HyperText Markup Language, es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTML se escribe en forma de etiquetas y también puede describir, hasta un cierto punto, la apariencia de un documento. Puede incluir uno o varios scripts, los cuales pueden afectar el comportamiento de navegadores web y otros procesadores de HTML.

¹⁰ <http://www.atinachile.cl/content/view/32755/En-que-consiste-la-Internet2.html>

1.2.1 Elementos

Los elementos son la estructura básica de HTML. Tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML.

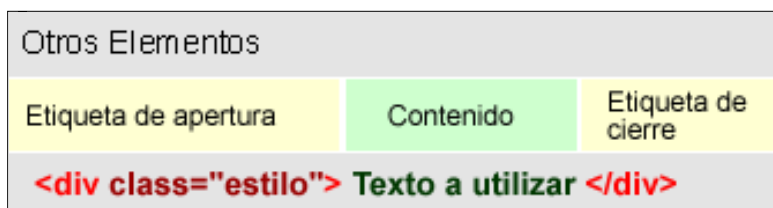


FIG 1-02 Estructura general de una etiqueta HTML [A]

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico y utiliza etiquetas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en el navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

1.2.2 Cronología del HTML

1986: Publicación de la ISO 8879 que presenta el Standard General Markup Language, origen del HTML.

1989: Tim Berners-Lee¹¹, presenta su artículo Information Management: A Proposal dedicándose de lleno al desarrollo de un sistema que permitiera el acceso en línea de manera uniforme a la información disponible en muchos recursos distintos, y que pudiese funcionar en máquinas que conectadas por redes basadas en TCP/IP.

1990 – 1991: Tim Berners-Lee define el HTML como un subconjunto de SGML¹². Se crea el nombre World Wide Web.

¹¹ Tim Berners Lee: Inventor de la www, padre de la web creador del sistema de localización de objetos en la web URL

¹² SGML: Standard Generalized Markup Language.

1991: Tim Berners-Lee introduce el primer visor de HTML, LineMode, que trabaja en modo texto y sólo en plataformas UNIX¹³.

1992: Se produce la primera Definición de Tipo de Documento (DTD) para el lenguaje, llamada HTML 1.0, agregando a la definición original atributos para modificar el estilo físico del texto.

1993: A fines de año, comienzan a aparecer los primeros artículos sobre WWW en diarios y revistas de circulación masiva.

1994: Se redefine el HTML para el nivel 2.0, que ahora soporta formularios. Un grupo de programadores desarrollan un nuevo visor de World Wide Web, Netscape. Se define un equivalente para los modelos en tres dimensiones del HTML, el VRML¹⁴, que permite moverse dentro de los ambientes definidos. En este mismo año se realizan la Primera y Segunda conferencias internacionales de WWW, en Ginebra y Chicago, respectivamente. Se crea la W3C¹⁵ Organization.

1995: Dave S. Raggett comienza a compilar la normativa del nuevo nivel del lenguaje, el HTML 3.0, cuya principal novedad es el soporte de tablas. Microsoft produce su primer visor de Internet, el cual también utiliza elementos de HTML degenerados. Una nueva versión de Netscape, Navigator 2.0, agrega soporte de encuadres. Sun Microsystems produce el primer visor de World Wide Web con soporte de un lenguaje de programación, HotJava.

1996: Netscape Communications y Microsoft presentan las nuevas versiones de sus visores que soportan gran parte del nivel de HTML 3.0. Aparecen visores no comerciales que implementan la norma completa de HTML 3.0. Se formaliza un nuevo nivel para la modelación en tres dimensiones, VRML 3.0, que permite interactuar con los objetos definidos.

¹³ Unix: Sistema Operativo

¹⁴ VRML: Virtual Real Modelity Lenguaje.

¹⁵ W3C: Consorcio internacional que produce recomendaciones para la World Wide Web.

1997: D. Raggett presenta, la versión normalizada del 3.2.

1998. Aparece el HTML 4.1 y a partir de aquí se definen nuevo estándares en css¹⁶, xml¹⁷, html y xhtml gracias a la W3C.

2010 - 2011: Se está terminando el desarrollo de HTML 5, la cual establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos.

1.2.3 FUNCIONAMIENTO DE LA WEB

El primer paso consiste en traducir la parte nombre del servidor de la URL¹⁸ en una dirección IP usando la base de datos distribuida de Internet conocida como DNS¹⁹. Esta dirección IP es necesaria para contactar con el servidor web y poder enviarle paquetes de datos.

El siguiente paso es enviar una petición HTTP²⁰ al servidor Web solicitando el recurso. En el caso de una página web típica, primero se solicita el texto HTML y luego es inmediatamente analizado por el navegador, el cual, después, hace peticiones adicionales para los gráficos y otros ficheros que formen parte de la página.

Al recibir los ficheros solicitados desde el servidor web, el navegador renderiza la página tal y como se describe en el código HTML, el css y otros lenguajes web. Al final se incorporan las imágenes y otros recursos para producir la página que ve el usuario en su pantalla.

Si un usuario accede de nuevo a una página después de un pequeño intervalo, es probable que no se vuelvan a recuperar los datos del servidor web de la forma en que se explicó en el apartado anterior. Por defecto, los navegadores almacenan en una caché del disco duro local todos los recursos web a los que el usuario va accediendo. El navegador enviará una petición HTTP sólo si la página ha sido actualizada desde la última carga, en otro caso, la versión

¹⁶ CSS: Cascading Style Sheet.

¹⁷ XML: Extensible Markup Language.

¹⁸ URL: Uniform Resource Locator.

¹⁹ DNS: Domain Name System.

²⁰ HTTP: Hypertext Transfer Protocol.

almacenada se reutilizará en el paso de renderizado para agilizar la visualización de la página. Esto es particularmente importante para reducir la cantidad de tráfico web en Internet.

Aparte de las utilidades creadas en los servidores Web que pueden determinar cuándo los ficheros físicos han sido actualizados, los diseñadores de páginas web generadas dinámicamente pueden controlar las cabeceras HTTP enviadas a los usuarios, de forma que las páginas intermedias o sensibles a problemas de seguridad no sean guardadas en caché.

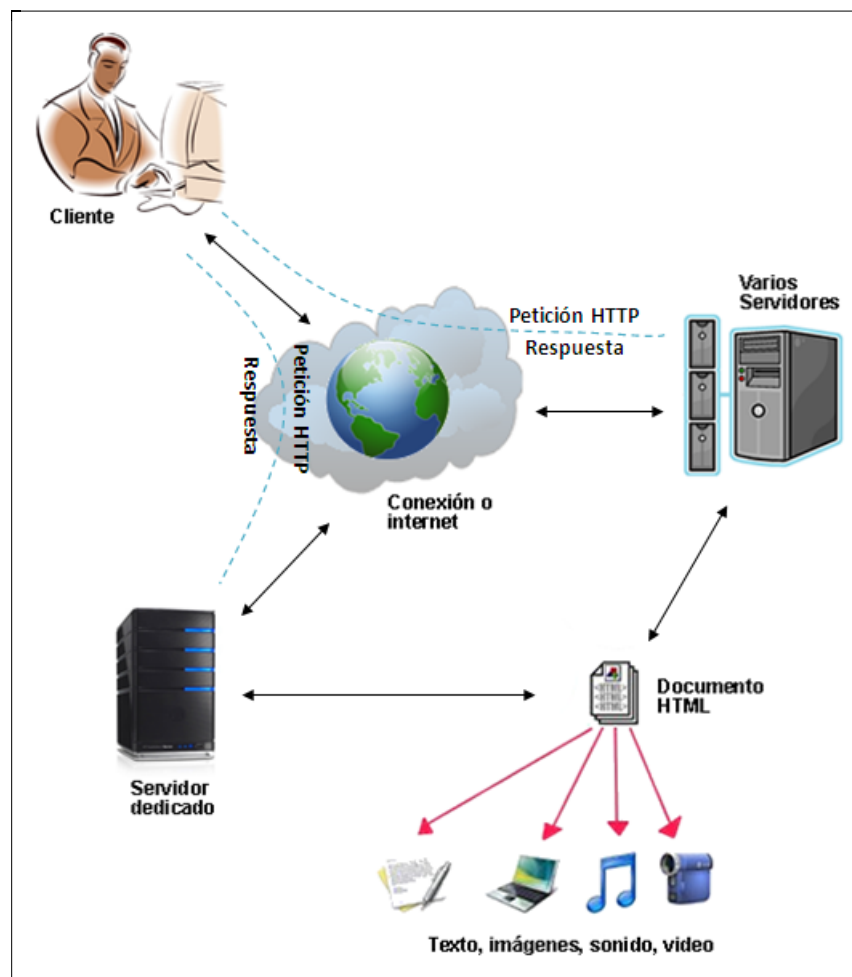


FIG 1-03 Diagrama de funcionamiento de la Web [A]

1.3. SOFTWARE LIBRE

Decimos que el software libre se refiere a la libertad para que los usuarios puedan ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a software gratuito.

No hay que confundir el software libre con el software de dominio público, ya que éste último es aquel software que no requiere de licencia.

1.3.1 Libertades del software libre

De acuerdo con tal definición, el software es libre si garantiza las siguientes libertades:

Libertad	Descripción
0	Usar el programa con cualquier propósito.
1	Estudiar cómo funciona el programa y modificarlo, adaptándolo a las necesidades que se deseen.
2	Distribuir copias del programa con lo cual se puede ayudar a los demás
3	Mejorar el programa y hacer públicas esas mejoras para los demás, de modo que todo el mundo se beneficie.

TABLA 1-01 Libertades del Software Libre [1]

En la definición de software libre no se contempla el asunto del precio. El software gratis puede incluir restricciones que no se adaptan a la definición de software libre por ejemplo, puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio, etc.

1.3.2 TIPOS DE LICENCIAS

Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer actos de explotación legales. Existen distintas variantes del concepto o grupos de licencias:

1.3.2.1 *Licencias GPL*

Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). Con este tipo de licencia el autor conserva los derechos de autor, y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Es decir, se posibilita la modificación y redistribución del software, pero únicamente bajo esta misma licencia.

1.3.2.2 *Licencias AGPL*

La Licencia Pública General de Affero es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

Es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

1.3.2.3 *Licencias estilo BSD*

Se les llama así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Puede argumentarse que esta licencia asegura verdadero software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre.

1.3.2.4 Licencias estilo MPL y derivadas

Esta licencia es de Software Libre y promueve eficazmente la colaboración evitando el efecto viral de la GPL. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

1.3.2.5 Copyleft

Hay que hacer constar que el titular de los derechos de autor (copyright²¹) de un software bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. En términos legales, el autor no tiene derecho a retirar el permiso de una licencia en vigencia. Si así sucediera, el conflicto entre las partes se resolvería en un pleito convencional.

1.3.3 Comparación con el software de código abierto

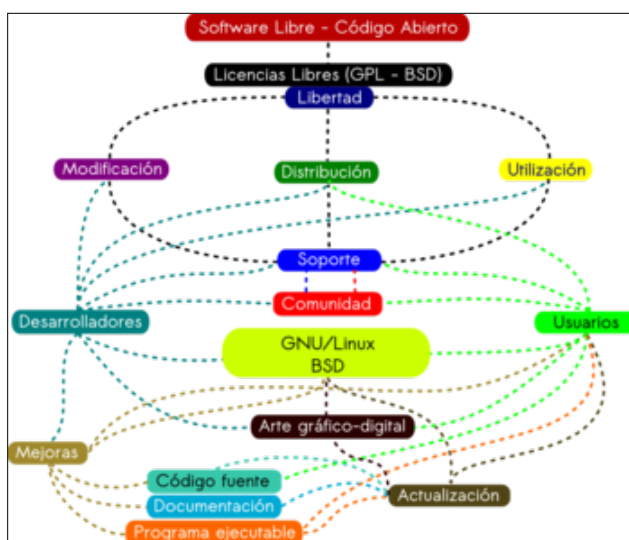


FIG 1-04 Mapa conceptual del software libre y de código abierto [1]

²¹ Copyright: Derechos de Autor.

1.3.4 Ventajas del software libre

- Bajo costo de adquisición.
- Innovación tecnológica: Esto se debe a que cada usuario puede aportar sus conocimientos y su experiencia.
- Independencia del proveedor.
- Escrutinio público (Corrección de Errores).
- Adaptación del software.

1.3.5 Impacto del software libre

Los impactos del software libre, y las principales nuevas perspectivas que permite, son los siguientes:

- Aprovechamiento más adecuado de los recursos: Muchas aplicaciones utilizadas o promovidas por las administraciones públicas son también utilizadas por otros sectores de la sociedad.
- Fomento de la industria local: Una de las mayores ventajas del software libre es la posibilidad de desarrollar industria local de software.
- Independencia del proveedor: Una organización preferirá depender de un mercado en régimen de competencia que de un solo proveedor que puede imponer las condiciones en que proporciona su producto.
- Adaptación a las necesidades exactas: En el caso del software libre, la adaptación puede hacerse con mucha mayor facilidad, y lo que es más importante, sirviéndose de un mercado con competencia, si hace falta contratarla.
- Escrutinio público de seguridad: Para una administración pública poder garantizar que sus sistemas informáticos hacen sólo lo que está previsto que hagan es un requisito fundamental y, en muchos estados, un requisito legal.
- Disponibilidad a largo plazo: Muchos datos que manejan las administraciones y los programas que sirven para calcularlos han de estar disponibles dentro de decenas de años.

1.4. PHP

Hypertext Pre-Processor es un lenguaje de programación interpretado de alto nivel, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor, tomando el código en PHP como su entrada y creando páginas web como salida, pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Fue creado originalmente por Rasmus Lerdorf²² en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation²³ considera esta licencia como software libre.

Por otro lado PHP es un lenguaje de propósito general ampliamente usado, que puede ser incrustado dentro de código HTML. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. La versión más reciente de PHP es la 5.3.5, la cual apareció el 6 de enero del 2011 solventando un error en la conversión de string a entero.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica, por ejemplo obteniendo información de una base de datos. El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

²² Rasmus Lerdorf: Creador de la primera versión del lenguaje Php.

²³ Free Software Foundation: Organización creada con el propósito de difundir el software libre.

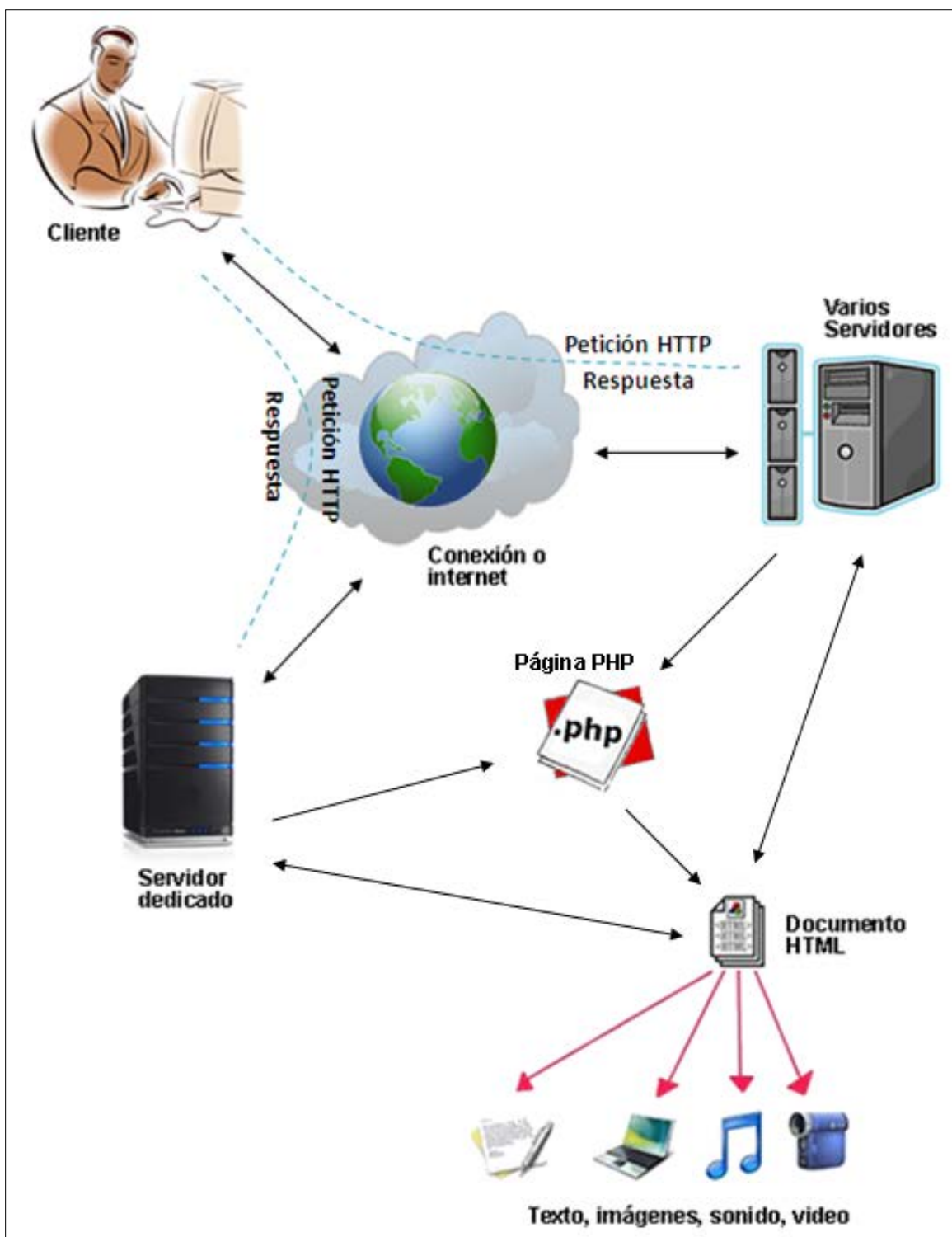


FIG 1-05 Esquema del funcionamiento de las páginas PHP [A]

PHP permite la conexión a diferentes tipos de gestores de bases de datos tales como: MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite, Adabas D, dbm, dBase, filePro, Hyperwave, Informix, InterBase, LDAP, mSQL, MySQL, Solid, Sybase.

Un servidor web compatible con PHP es XAMPP. Éste es un servidor independiente de plataforma, es software libre, y se relaciona principalmente con la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X - para cualquiera de los diferentes sistemas operativos -. El programa está liberado bajo la licencia GNU, es fácil de usar y capaz de interpretar páginas dinámicas.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos. Entre otras características PHP ofrece soporte para: Acceso a servidores IMAP²⁴, Envío de correo con SMTP²⁵, Acceso a servidores de FTP²⁶, Acceso a SNMP²⁷ para gestión de redes y equipos, Generación dinámica de gráficos y documentos PDF, Análisis de documentos XML, Corrector de ortografía.

Todo esto ha sido posible gracias a que PHP ha sido diseñado de forma muy modular lo que permite la incorporación de nuevas librerías.

1.4.1 Ventajas

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- Posee una amplia documentación en su página.

²⁴ IMAP: Internet Message Access Protocol.

²⁵ SMTP: Simple Mail Transfer Protocol.

²⁶ FTP: File Transfer Protocol.

²⁷ SNMP: Simple Network Management Protocol.

- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable.
- PHP, en el caso de estar montado sobre un servidor Linux u Unix, se ejecuta en un único espacio de memoria haciendo que su rapidez sea mayor.
- PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades.

1.4.2 Las cuatro grandes características de PHP

Velocidad: No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix.

Estabilidad: La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de errores, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para los errores sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.



FIG 1-06 Características de PHP [A]

Seguridad: El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo `php.ini`²⁸

Simplicidad: Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

1.4.3 Inconvenientes

- La ofuscación de código es la única forma de ocultar las fuentes.

1.5. AJAX

Esta técnica de desarrollo web, llamada Asynchronous JavaScript And XML fue desarrollada para crear aplicaciones interactivas o RIA²⁹. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en

²⁸ `Php.ini`: Archivo de configuración del ambiente de Php y otras características.

²⁹ RIA: Rich Internet Applications.

segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

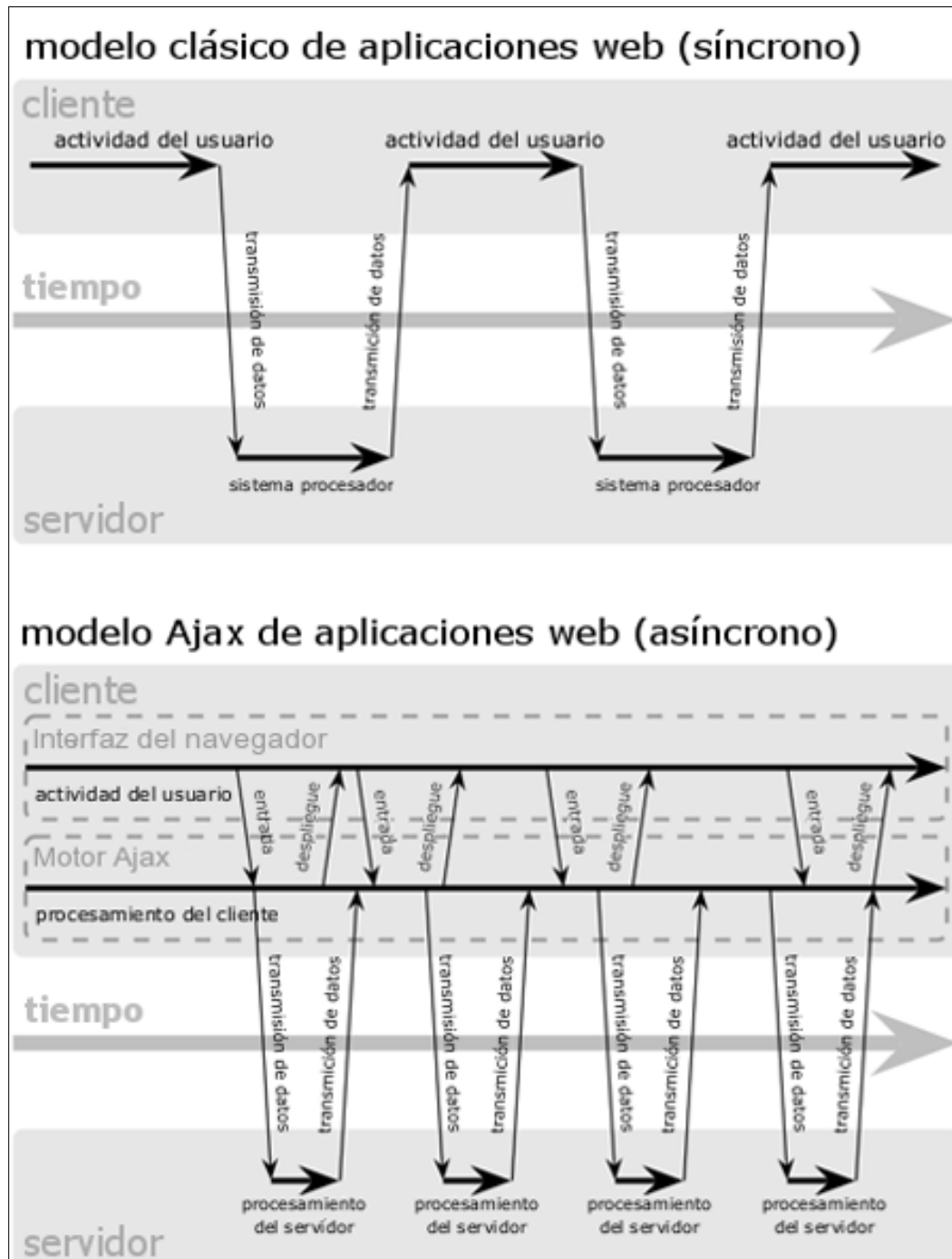


FIG 1-07 Modelo AJAX de aplicaciones Web [2]

JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest³⁰, objeto disponible en los navegadores actuales.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DOM.

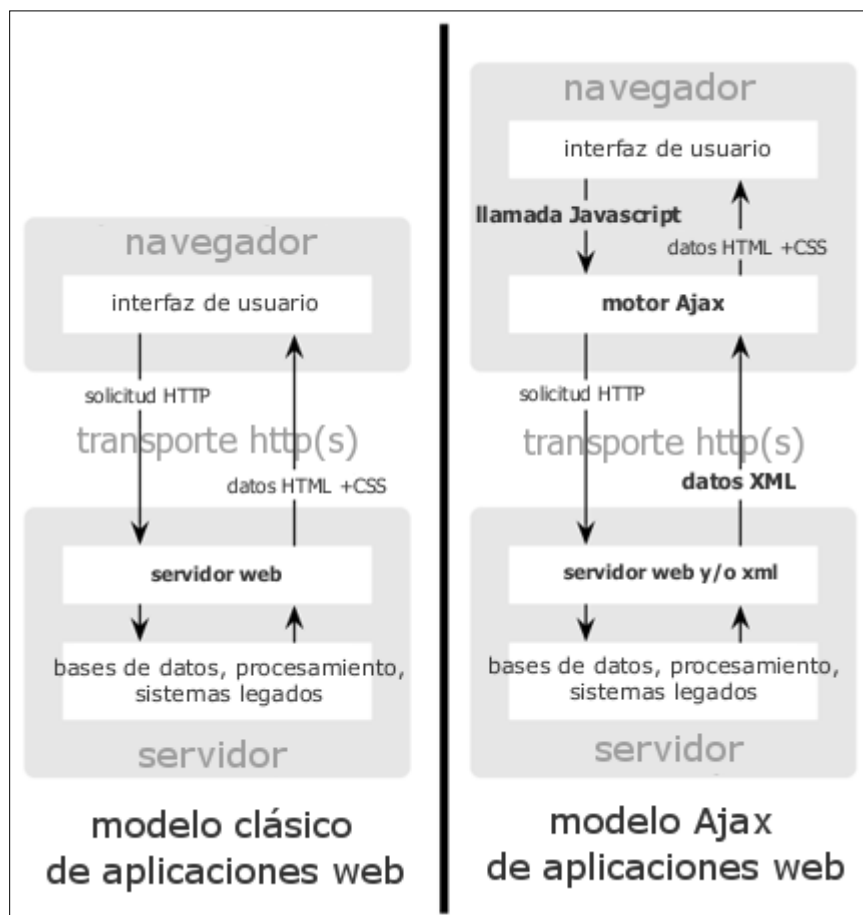


FIG 1-08 Modelo AJAX de aplicaciones Web II [2]

Ajax es una combinación de cuatro tecnologías ya existentes:

- XHTML o HTML y hojas de estilos en cascada para el diseño que acompaña a la información.

³⁰ XMLHttpRequest: Extensible Markup Language / Hypertext Transfer Protocol.

- Document Object Model accedido con un lenguaje de scripting por parte del usuario.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, etc.

Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

1.5.1 Ventajas

- Basado en los estándares abiertos (Javascript, html, xml, css, y XML HTTP Request.
- Usabilidad (No hace falta recargar la página entera).
- Válido en cualquier plataforma y navegador actual.
- Beneficia las aplicaciones web.
- No es difícil su utilización.
- Compatible con Flash.
- Se basa en el concepto de Web 2.0
- Es independiente del tipo de tecnología de servidor que se utilice
- Mejora la estética de la web.
- Es compatible con Flash

1.5.2 Problemas e Inconvenientes

- Las páginas con Ajax son más difíciles de desarrollar que las páginas estáticas.
- Las páginas creadas dinámicamente mediante peticiones sucesivas Ajax, no son registradas de forma automática en el historial del navegador.
- Los motores de búsquedas no entienden JavaScript, por lo que la información en la página dinámica no se almacena en los registros del buscador.
- Hay problemas usando Ajax entre nombres de dominios. Eso es una función de seguridad.
- El sitio con Ajax usa más recursos en el servidor.
- De momento no funcionan todas las funcionalidades en dispositivos móviles.

1.6. MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario, que es desde abril de 2009 parte de Oracle Corporation, el cual desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para las empresas que quieran incorporarlo en productos privativos deben comprar una licencia específica que permita hacer uso este. Está desarrollado en su mayor parte en ANSI C.

MySQL es muy utilizado para el desarrollo de aplicaciones web, en Drupal³¹ o PHP. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

1.6.1 Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

³¹ Drupal: Gestor de contenidos o CMS

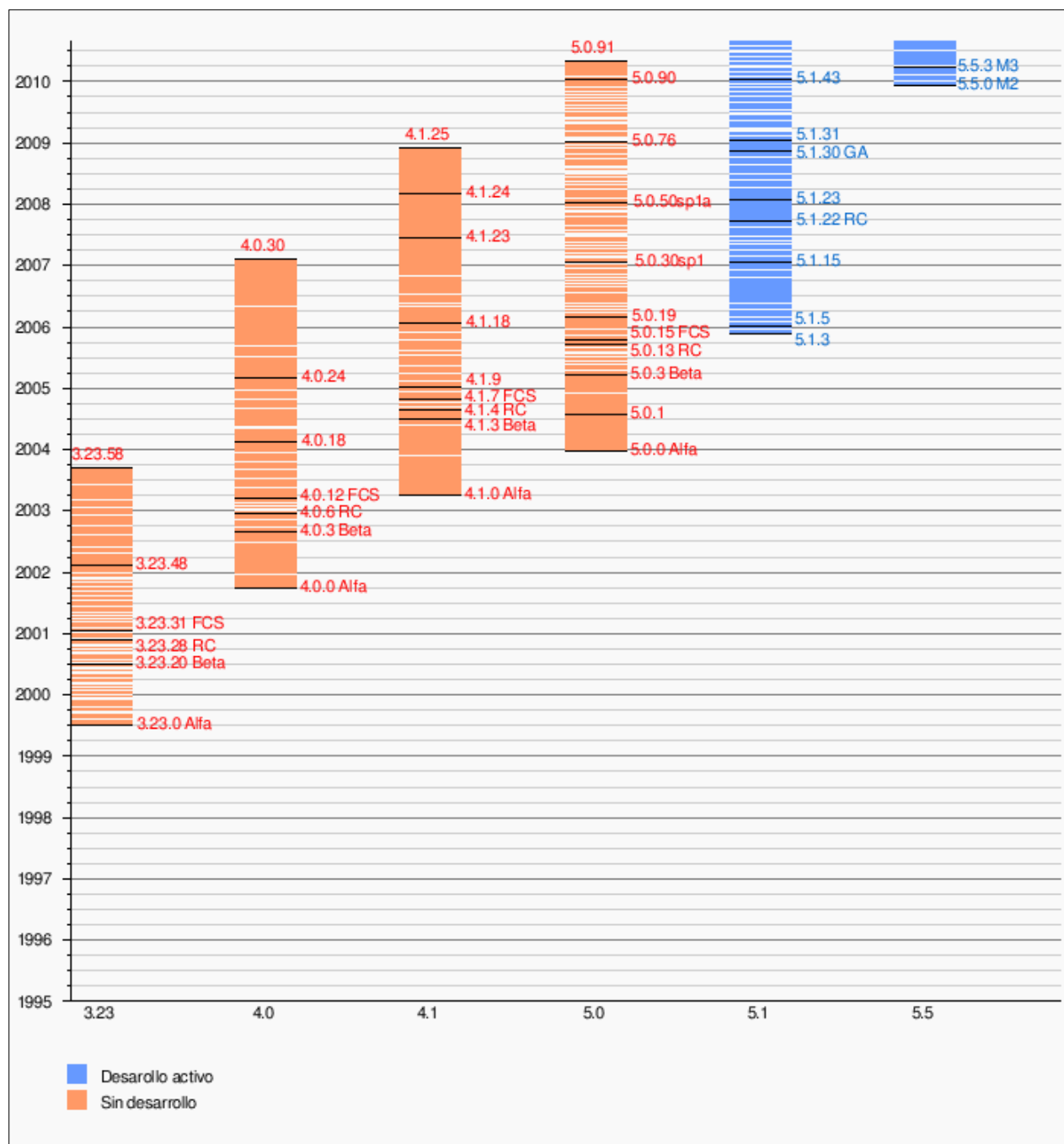


FIG 1-09 Evolución y versionamiento de MySQL [3]

1.6.2 Tipos de compilación del servidor

Hay tres tipos de compilación del servidor MySQL:

- Estándar: Los binarios estándares de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max: Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL - Debug: Son binarios que han sido compilados con información de depuración extra. No se debe usar en sistemas en producción porque el código de depuración puede reducir el rendimiento.

1.7. CLOUD COMPUTING: LA NUEVA ERA DEL DESARROLLO

La computación en la nube, es un paradigma que permite ofrecer servicios de computación a través de Internet, en donde los usuarios de este servicio tienen acceso de forma gratuita o de pago al servicio que necesite usar. El término es una tendencia que responde a múltiples características integradas.

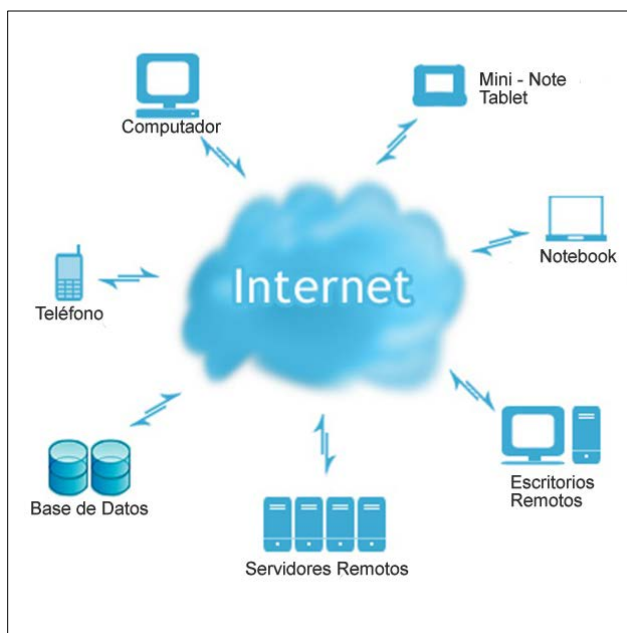


FIG 1-10 Esquema principal de Cloud Computing [4]

1.7.1 INTRODUCCIÓN

En este tipo de metodología, todo lo que puede ofrecer un sistema informático se ofrece como servicio, (servicios web) de modo que los usuarios puedan acceder a los servicios disponibles en el Internet sin conocimientos (o, al menos sin ser expertos) en la gestión de los recursos que usan. Muchas entidades dicen que es un paradigma en el que la información se almacena de manera permanente en servidores de Internet y se envía a cachés temporales de cliente, lo que incluye equipos de escritorio, centros de ocio, portátiles, etc. Esto se debe a que, pese a que las capacidades de los PC han mejorado sustancialmente, gran parte de su potencia se desaprovecha, al ser máquinas de propósito general.

Cloud Computing es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado.

El cambio paradigmático que ofrece computación en nube genera beneficios tanto para los proveedores, que pueden ofrecer, de forma más rápida y eficiente, un mayor número de servicios, como para los usuarios que tienen la posibilidad de acceder a ellos, disfrutando de la transparencia e inmediatez del sistema y de un modelo de pago por consumo.

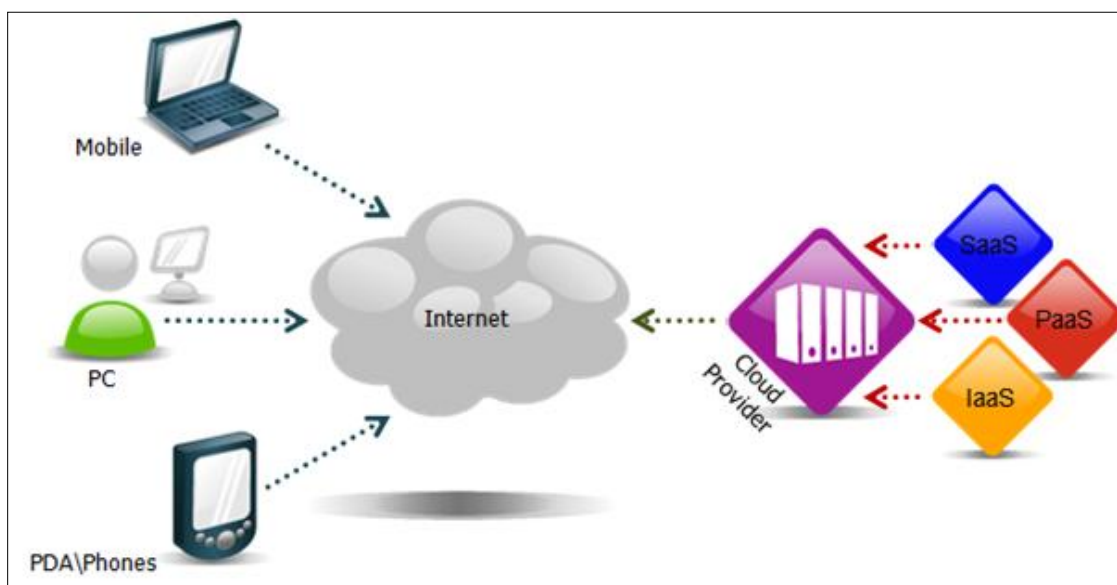


FIG 1-11 Cloud Computing orientado al servicio [5]

La computación en nube es un concepto que incorpora el software como servicio, como en la Web 2.0 y otros conceptos recientes, también conocidos como tendencias tecnológicas, que tienen en común el que confían en Internet para satisfacer las necesidades de cómputo de los usuarios.

1.7.2 CARACTERÍSTICAS DEL CLOUD COMPUTING

Una de las principales características del Cloud Computing es que no hay necesidad de conocer la infraestructura detrás de esta ya que pasa a ser un lugar donde las aplicaciones y servicios pueden fácilmente crecer, funcionar rápido y casi nunca fallan, sin conocer los detalles del funcionamiento de ésta.

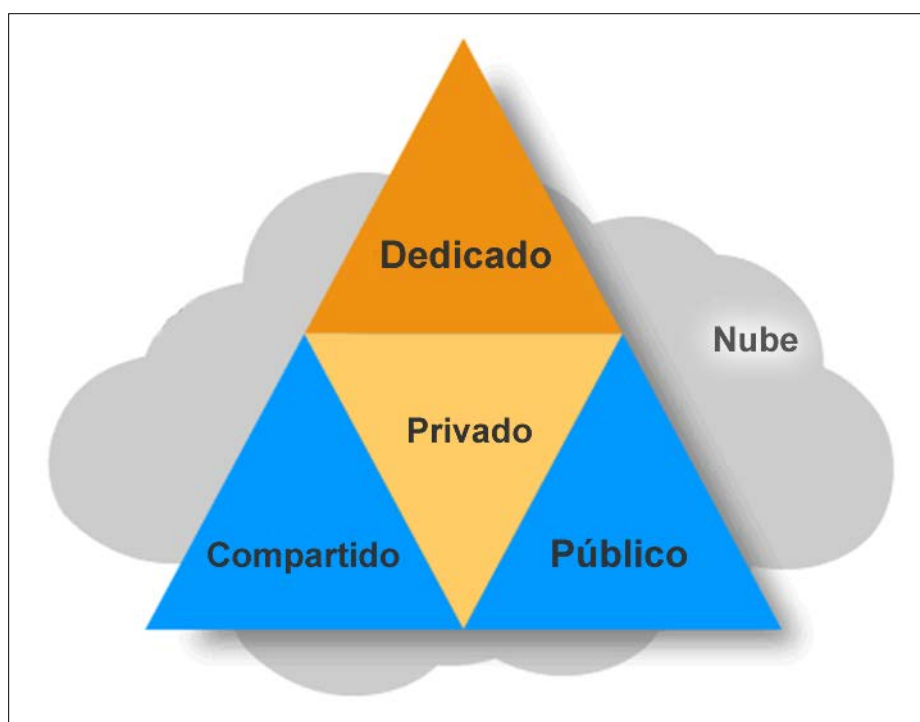


FIG 1-12 Evolución de Cloud Computing [6]

Entre otras características podemos mencionar:

- Auto Reparable: En caso de fallo, el último backup de la aplicación pasa a ser automáticamente la copia primaria y se genera uno nuevo.

- Escalable: Todo el sistema/arquitectura es predecible y eficiente. Si un servidor maneja 1000 transacciones, 2 servidores manejarán 2000 transacciones.
- Regidos por un Acuerdo de Nivel de Servicio (SLA) que define varias políticas como cuales son los tiempos esperados de rendimiento y en caso de pico, debe crear más instancias.
- Virtualizado: las aplicaciones son independientes del hardware en el que corran, incluso varias aplicaciones pueden correr en una misma máquina o una aplicación puede usar varias máquinas a la vez.
- Multipropósito: El sistema está creado de tal forma que permite a diferentes clientes compartir la infraestructura sin preocuparse de ello y sin comprometer su seguridad y privacidad.

1.7.3 VENTAJAS

Algunas de las ventajas de la computación en las nubes son:

- Acceso a la información y los servicios desde cualquier lugar.
- Servicios gratuitos y de pago según las necesidades del usuario.
- Empresas con facilidad de escalabilidad
- Capacidad de procesamiento y almacenamiento sin instalar máquinas localmente.
- Integración probada de servicios Red: Cloud Computing se puede integrar con mucha mayor facilidad y rapidez con el resto de sus aplicaciones empresariales, independientemente del software, y a pesar de que sean desarrolladas de manera interna o externa.
- Prestación de servicios a nivel mundial: Las infraestructuras de Cloud Computing proporcionan mayor capacidad de adaptación, recuperación de desastres completa y reducción al mínimo de los tiempos de inactividad.
- Con Cloud Computing no se necesita instalar ningún tipo de hardware. Además de que requiere menor inversión para empezar a trabajar.
- Implementación más rápida y con menos riesgos. Las aplicaciones están disponibles en cuestión de semanas o meses, incluso con un nivel considerable de personalización o integración.

- Actualizaciones automáticas que no afectan negativamente a los recursos de TI³².
- Contribuye al uso eficiente de la energía. En este caso, a la energía requerida para el funcionamiento de la infraestructura.

1.7.4 DESVENTAJAS

- La centralización de las aplicaciones y el almacenamiento de los datos origina una dependencia de los proveedores de servicios.
- La disponibilidad de las aplicaciones están atadas a la disponibilidad de acceso a internet.
- Los datos sensibles del negocio no residen en las instalaciones de las empresas por lo que podría generar un contexto de alta vulnerabilidad para la sustracción o robo de información.
- La disponibilidad de servicios altamente especializados podría tardar meses o incluso años para que sean factibles de ser desplegados en la red.
- La madurez funcional de las aplicaciones hace que continuamente estén modificando sus interfaces por lo cual la curva de aprendizaje en empresas de orientación no tecnológica tenga unas pendientes pequeñas.
- Con respecto a la seguridad, la información de la empresa debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos son un foco de inseguridad. Si se utilizan protocolos seguros, HTTPS por ejemplo, la velocidad total disminuye debido a la sobrecarga que requieren estos protocolos.
- Escalabilidad a largo plazo. A medida que más usuarios empiecen a compartir la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentaran, si la empresa no posee un esquema de crecimiento óptimo puede llevar a degradaciones en el servicio altos.

³² IT: Technology Information ó Tecnología de la Información.

1.7.5 CAPAS

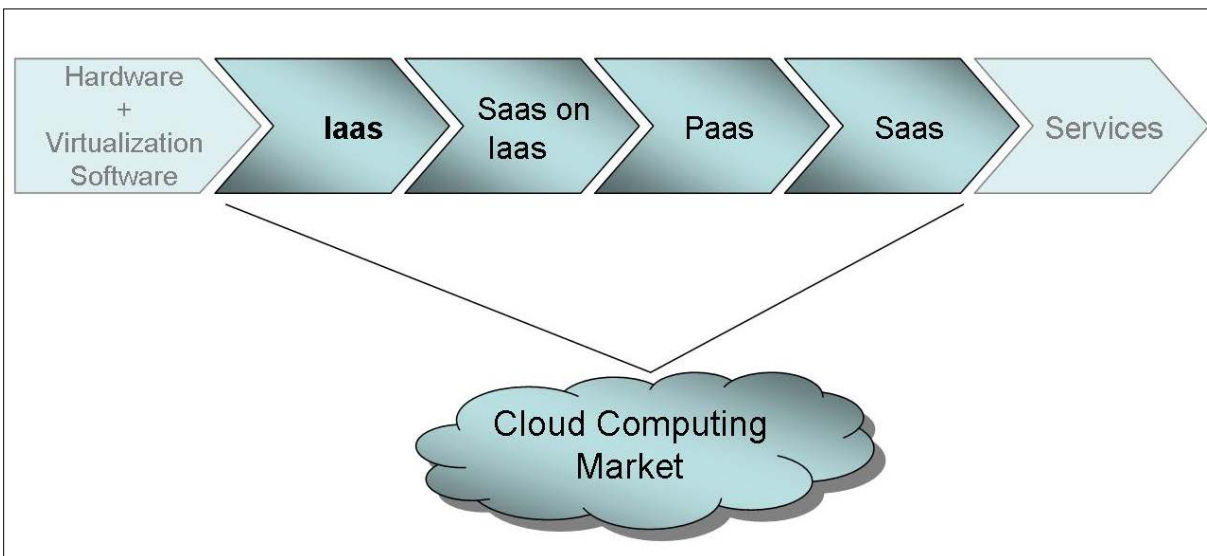


FIG 1-13 Infraestructura como un servicio [7]

1.7.5.1 Software como un Servicio (SaaS)

SaaS se encuentra en la capa más alta y caracteriza una aplicación completa ofrecida como un servicio, en demanda, vía multitenencia, que significa una sola instancia del software que corre en la infraestructura del proveedor y sirve a múltiples organizaciones de clientes.

1.7.5.2 Plataforma como un Servicio (PaaS)

La capa del medio, o PaaS, es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una carga de servicios.

1.7.5.3 Infraestructura como un Servicio (IaaS).

IaaS se encuentra en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red. Servidores, sistemas de almacenamiento, conexiones, enrutadores, y otros sistemas se concentran, para manejar tipos específicos de cargas de trabajo; desde procesamiento en lotes (batch) hasta aumento de servidor/almacenamiento durante las cargas pico.

1.7.6 TIPOS DE NUBES

1. Las nubes públicas se manejan por terceras partes, y los trabajos de muchos clientes diferentes pueden estar mezclados en los servidores, los sistemas de almacenamiento y otras infraestructuras de la nube. Los usuarios finales no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, discos como los suyos propios.
2. Las nubes privadas son una buena opción para las compañías que necesitan alta protección de datos y ediciones a nivel de servicio. Las nubes privadas están en una infraestructura en-demanda manejada por un solo cliente que controla qué aplicaciones debe correr y dónde. Son propietarios del servidor, red, y disco y pueden decidir qué usuarios están autorizados a utilizar la infraestructura.
3. Las nubes híbridas combinan los modelos de nubes públicas y privadas. Usted es propietario de unas partes y comparte otras, aunque de una manera controlada. Las nubes híbridas ofrecen la promesa del escalado aprovisionada externamente, en demanda, pero añaden la complejidad de determinar cómo distribuir las aplicaciones a través de estos ambientes diferentes. Las empresas pueden sentir cierta atracción por la promesa de una nube híbrida, pero esta opción, al menos inicialmente, estará probablemente reservada a aplicaciones simples sin condicionantes, que no requieran de ninguna sincronización o necesiten bases de datos complejas.

1.7.7 EJEMPLOS DE CLOUD COMPUTING

- Google Apps: brinda el servicio de aplicaciones para empresas como: Gmail, Google Talk, Google Calendar y Google Docs, etc.
- Amazon Web Services: los servicios que ofrece son el Amazon EC2™, Amazon S3™, SimpleDB™, Amazon SQS™.
- Azure de Microsoft: ofrece servicios de sistema operativo, hosting, sistemas para desarrollo.
- Rackspace: una famosa empresa de hosting, está detrás de Mosso una compañía la cual ofrece un paquete básico de hosting y a partir de este se escala según se necesite.

1.8. ARQUITECTURA ORIENTADA A SERVICIOS (SOA)

La urgencia de datos, la exactitud y la seguridad a partir de un proceso de negocio a otro, son ahora una prioridad de negocio. Las organizaciones que pueden hacer esto tienen una distintiva ventaja competitiva. El objetivo de contar con un sistema unificado, significa para las compañías estar preparadas para cambiar sus focos y recursos desde el mantenimiento de las aplicaciones separadas al desarrollo de procesos de negocios de punta a punta basados en el servicio al cliente.

Asimismo, la flexibilidad de un sistema unificado hace esto posible para cambiar aquellos procesos en una respuesta rápida a los cambios en las necesidades de los negocios.

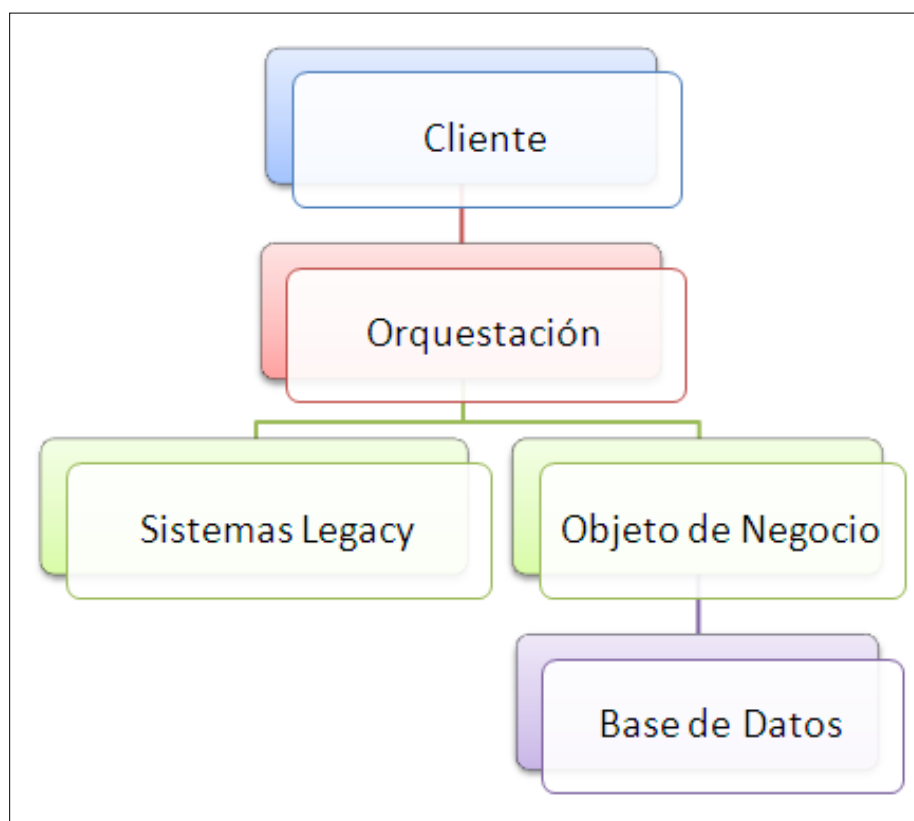


FIG 1-14 Estructura SOA [A]

La arquitectura orientada a servicios no se trata de software o de un lenguaje de programación, SOA es un marco de trabajo conceptual que permite a las organizaciones unir los objetivos de negocio con

la infraestructura de TI integrando los datos y la lógica de negocio de sus sistemas separados. Es decir es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización y a su vez brinda una forma bien definida de exposición e invocación de servicios, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

SOA define las siguientes capas de software:

- Aplicaciones básicas: Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.
- De exposición de funcionalidades: Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios.
- De integración de servicios: Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.
- De composición de procesos: Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio,
- De entrega:- Donde los servicios son desplegados a los usuarios finales.

Con SOA, una empresa puede mantener sus inversiones en los sistemas legacy y la gente necesaria para mantenerlos. Esto evita continuos y costosos proyectos "de integración", como las mejoras a cualquier aplicación son transparentes a todas las otras. La información de negocio está siempre actualizada y en línea, permitiendo mejores decisiones de negocio y mejorando las relaciones entre clientes y partners.

1.8.1 HISTORIA

SOA se Desarrolló a finales de los años 90, y establece un marco de trabajo para servicios de red; o tareas comunes de negocios; para identificar el uno al otro y comunicarlo.

En los comienzos, los desarrollos de aplicaciones de negocio se concentraban en necesidades específicas: contabilidad, compras, nómina de sueldos, transporte. Cada aplicación fue desarrollada sin consideración de otros sistemas en la empresa y como comunicarse con ellos. Porque las aplicaciones eran auto suficientes, la información común a

toda la empresa (la dirección del cliente) y funciones específicas de negocios (buscar un nombre) aparecían en todas partes y requerían un código complejo para, todos o muchos de los sistemas independientes.

Por consiguiente, los diversos sistemas de TI de la mayoría de las empresas hoy no pueden acceder o procesar los datos desde el uno al otro. Un simple proceso de negocio – como una venta para un pedido a un depósito enviado a una cuenta por cobrar – que tomaría segundos si los sistemas se podrían comunicar, ahora puede tomar semanas.

1.8.2 BENEFICIOS

Los beneficios que puede obtener una organización que adopte SOA son:

- Mejora en los tiempos de realización de cambios en procesos.
- Facilidad para evolucionar a modelos de negocios basados en tercerización.
- Facilidad para abordar modelos de negocios basados en colaboración con otros entes.
- Poder para reemplazar elementos de la capa applicativa SOA sin interrupción en el proceso de negocio.
- Facilidad para la integración de tecnologías disímiles.

1.8.3 DESVENTAJA

Como toda tecnología, los servicios web pueden ser mal implementados, si no se siguen los estándares y buenas prácticas, que existen para el desarrollo de Servicios. Uno de los problemas más comunes es pretender que toda funcionalidad de un sistema se puede convertir en un servicio web, y la verdad es que el enfoque es otro, las funcionalidades de Negocio, que pueden involucrar más de un sistema, son las que se deben convertir en Servicios.

1.9. METODOLOGÍAS ÁGILES

En el 2001 en Utah – EEUU, se realizó una reunión que comprendió expertos de la industria de software, así como algunos creadores e impulsores de metodologías de software, y cuyo objetivo principal fue esbozar los valores y principios que deberían permitir a los equipos, desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo de un proyecto, nació el término ágil.

Este término se acuñó ya que se pretende ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas Internamente.	Impuestas Externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos Pequeño y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos.
Pocos Artefactos.	Más Artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 1-02 Comparación entre Metodologías Ágiles y Metodologías Tradicionales [8]

Existen varias metodologías de programación consideradas ágiles pero según Kent Beck³³, Extreme Programming es el más destacado de los procesos ágiles de desarrollo de software.

	CMM	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Características Metodología (CM)								
- Resultado	2	5	5	4	4	4	5	5
- Simplicidad	1	4	4	3	5	3	5	5
- Adaptabilidad	2	5	5	3	3	4	4	3
- Excelencia Técnica	4	3	3	4	4	4	3	4
- Prácticas de Colaboración	2	5	5	4	3	3	4	5
Media CM	2,2	4,4	4,4	3,6	3,8	3,6	4,2	4,4
Media Total	1,7	4,8	4,5	3,6	3,6	3,9	4,7	4,8

Tabla 1-03 Ranking de Agilidad entre las diferentes metodologías [8]

1.9.1 PROGRAMACIÓN EXTREMA (EXTREME PROGRAMMING, XP)

Es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Este tipo de programación es la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con un proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Esta programación es adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

³³ Kent Beck: Uno de los creadores de la metodología ágil para el desarrollo de software conocida como programación extrema.

La programación extrema se diferencia de las metodologías tradicionales principalmente porque pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de este tipo de programación consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

1.9.2 VALORES

Algunos de los valores que se destacan en la Programación Extrema son:

1. **Simplicidad:** Es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hacen que la complejidad aumente exponencialmente. Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece.

También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando eso sí que el código esté autodocumentado. Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

2. **Comunicación:** Se la realiza de diferentes formas. Para los programadores el código comunica mejor cuanto más simple sea. El código autodocumentado es más fiable que los comentarios ya que éstos últimos pronto quedan desfasados con el código a medida que es modificado. Debe comentarse sólo aquello que no va a variar, por ejemplo el objetivo de una clase o la funcionalidad de un método. Las pruebas unitarias son otra forma de comunicación ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo utilizar su funcionalidad.

Los programadores se comunican constantemente gracias a la programación por parejas. La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo.

El cliente decide que características tienen prioridad y siempre debe estar disponible para solucionar dudas.

3. Retroalimentación (*feedback*): Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante.
4. Coraje o valentía: Para los gerentes la programación en parejas puede ser difícil de aceptar, porque les parece como si la productividad se fuese a reducir a la mitad ya que solo la mitad de los programadores está escribiendo código. Hay que ser valiente para confiar en que la programación por parejas beneficia la calidad del código sin repercutir negativamente en la productividad. La simplicidad es uno de los principios más difíciles de adoptar.

Se requiere coraje para implementar las características que el cliente quiere ahora sin caer en la tentación de optar por un enfoque más flexible que permita futuras modificaciones. No se debe emprender el desarrollo de grandes marcos de trabajo mientras el cliente espera. En ese tiempo el cliente no recibe noticias sobre los avances del proyecto y el equipo de desarrollo no recibe retroalimentación para saber si va en la dirección correcta. La forma de construir marcos de trabajo es mediante la refactorización del código en sucesivas aproximaciones.

5. Respeto: El respeto se manifiesta de varias formas. Los miembros del equipo se respetan los unos a otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros respetan su trabajo porque siempre están luchando por la alta calidad en el producto y buscando el diseño óptimo o más eficiente para la solución a través de la refactorización del código.

1.9.3 ROLES DEFINIDOS EN LA PROGRAMACIÓN EXTREMA

1.9.3.1 Programador

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

1.9.3.2 Cliente

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

1.9.3.3 Encargado de pruebas (Tester)

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

1.9.3.4 Encargado de seguimiento (Tracker)

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

1.9.3.5 Entrenador (Coach)

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

1.9.3.6 Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

1.9.3.7 Gestor (Big boss)

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

1.9.4 CARACTERÍSTICAS DEL PROCESO DE XP

Un proyecto de XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo.



FIG 1-15 Proceso de Desarrollo [A]

El ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

1.9.5 CICLO DE VIDA IDEAL DE XP:

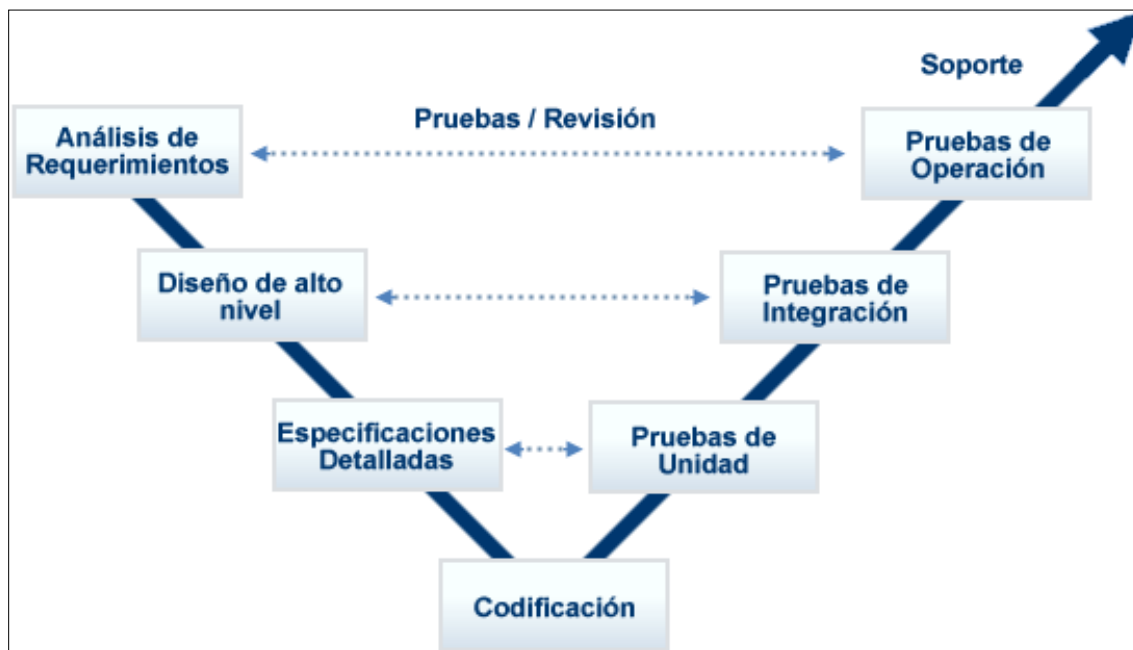


FIG 1-16 Ciclo de vida en V [A]

1.9.5.1 Fase I: Exploración

- Los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto.

- El equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.
- Esta fase toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

1.9.5.2 Fase II: Planificación de la Entrega

- El cliente establece la prioridad de cada historia de usuario.
- Los programadores realizan una estimación del esfuerzo necesario de cada una de las historias de los usuarios.
- Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.
- La primera entrega debería obtenerse en no más de tres meses.
- Esta fase dura unos pocos días y las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.
- El equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.
- La planificación se la realiza basándose en el tiempo o el alcance.

1.9.5.3 Fase III: Iteraciones

- Incluye varias iteraciones sobre el sistema antes de ser entregado.
- El Plan de Entrega está compuesto por iteraciones de no más de tres semanas.
- En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto.
- Se escogen las historias que fueren la creación de la arquitectura dicha en el punto anterior, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración.
- Cuando finalice la última iteración el sistema estará listo para entrar en producción.

- Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

1.9.5.4 Fase IV: Producción

- En esta fase se requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- Se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.
- Las ideas que han sido propuestas y las sugerencias, son documentadas para su posterior implementación.

1.9.5.5 Fase V: Mantenimiento

- Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.
- Se requiere de tareas de soporte para el cliente.
- La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

1.9.5.6 Fase VI: Muerte del Proyecto

- Se puede dar cuando:
- El cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura.
- El sistema no genera los beneficios esperados por el cliente.
- No hay presupuesto para mantenerlo.

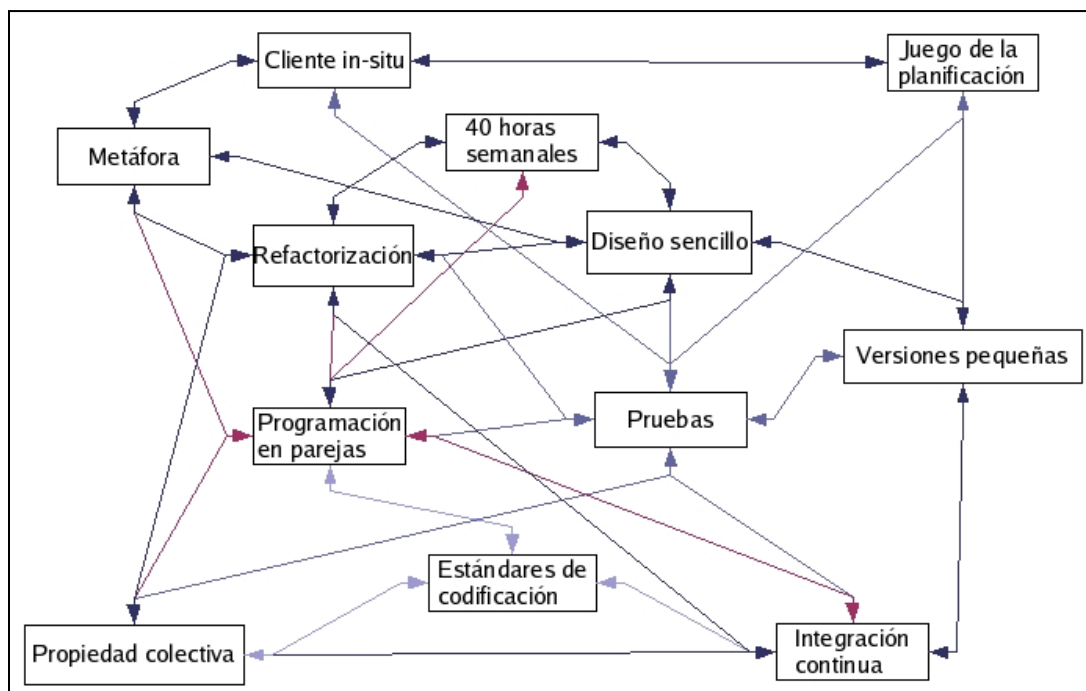


FIG 1-17 Las prácticas se refuerzan entre sí [8]

1.10. Contabilidad

Esta disciplina es una herramienta que se encarga de determinar, medir y cuantificar los factores de riqueza de las empresas, y que ayuda a conocer la situación financiera para poder tomar decisiones y tener un mejor control de ésta. Para ello se presenta información, previamente registrada y de manera sistémica, y luego se genera un balance.

En el balance se presenta todos los Estados Contables o Estados Financieros que son los que resumen la situación económica y financiera de la empresa. Esta información resulta útil para gestores, reguladores y otros tipos de interesados como los accionistas, acreedores o propietarios.

Si detallamos el Balance, vemos que éste se compone de dos columnas, el Activo y el Pasivo. El primero es el conjunto de cuentas que representan bienes y derechos tangibles e intangibles de la empresa o del ente económico y que, en la medida en que sean utilizados, pueden ser fuente potencial de beneficios presentes o futuros. Por otro lado el pasivo es el conjunto de cuentas que representa las obligaciones contraídas por la empresa o el ente económico en desarrollo del giro de su actividad, pagaderas en dinero, bienes o en servicios.

A parte del Activo y el Pasivo, en el balance también se presenta el patrimonio de la empresa o entidad económica. El patrimonio, también conocido como capital contable, agrupa el conjunto de las cuentas que representan el valor residual de comparar el activo total con el pasivo externo, producto de los recursos netos que han sido suministrados por el propietario de los mismos, ya sea directamente o como consecuencia del giro ordinario de sus negocios, es decir, por transacciones y otros eventos y circunstancias que afectan a la empresa. Comprende los aportes de los accionistas, socios o propietarios, entre otros aportes.

Existen varios tipos de contabilidad, pero según los tipos de usuarios (Externos-Internos), pueden ser:

1.10.1 CONTABILIDAD FINANCIERA.

Da la información esencial del funcionamiento y estado financiero de la empresa o ente económico a todos los agentes interesados, como por ejemplo: clientes, inversores, proveedores, Administraciones Públicas, etc.

1.10.2 CONTABILIDAD DIRECTIVA.

Se engloba la Contabilidad de costos o Contabilidad de gestión. Es la contabilidad interna, para el cálculo de los costos y movimientos económicos y productivos en el interior de la empresa. Sirve para tomar decisiones en cuanto a producción, organización de la empresa, etc.

1.10.3 CONTABILIDAD BÁSICA.

Aquí se presentan los estados y movimientos de las cuentas, en base al conjunto de activos y pasivos. No tiene mucho detalle el balance financiero, pero se presenta además de los activos y pasivos explícitos el estado de pérdidas y ganancias. Esto se lo hace en base a un conjunto de reglas establecidas por la entidad.

En esta disertación de grado nos vamos a enfocar principalmente en la CONTABILIDAD BÁSICA.

CAPÍTULO 2: ANÁLISIS Y DESARROLLO

2.1. IDENTIFICACIÓN DE REQUERIMIENTOS

A continuación se presentan los requerimientos funcionales y los requerimientos no funcionales que enmarcan el sistema contable en plataforma web. Estos requerimientos se basan principalmente en las necesidades prioritarias de los diferentes clientes que se podrían tener.

2.1.1 REQUERIMIENTOS FUNCIONALES

F0: Ingreso al Sistema

F1: Administración para el manejo de Usuarios

F1.1: El sistema será capaz de realizar el ingreso de nuevos usuarios.

F1.2: El sistema será capaz de realizar la actualización de usuarios.

F1.3: El sistema será capaz de realizar la eliminación de usuarios.

F2: Administración para el manejo de Cuentas Contables.

F2.1: El sistema permitirá realizar el manejo de ingresos de nuevas cuentas contables.

F2.2: El sistema permitirá realizar el manejo para la actualización de cuentas contables.

F2.3: El sistema permitirá realizar el manejo para la eliminación de cuentas contables.

F3: Administración para el manejo de Periodos.

F3.1: El sistema permitirá realizar el manejo para la actualización de los periodos contables.

F4: Administración para el manejo de Asientos Contables

F4.1: El sistema permitirá realizar el ingreso de asientos contables.

F4.2: El sistema permitirá realizar la actualización de asientos contables.

F4.3: El sistema permitirá realizar la eliminación de asientos contables.

F4.4: El sistema permitirá realizar y ejecutar el proceso de cuadro de valores.

F5: Procesos prioritarios del Sistema Contable

F5.1: El sistema permitirá realizar y ejecutar el proceso de mayorización.

F5.2: El sistema permitirá realizar y ejecutar el proceso de cierre de mes.

F5.3: El sistema permitirá realizar y ejecutar el proceso de cierre anual.

F6: Reportes

F6.1: El sistema permitirá mostrar un reporte del balance mensual.

F6.2: El sistema permitirá mostrar un reporte del balance anual.

F7: Salir del Sistema

2.1.2 REQUERIMIENTOS NO FUNCIONALES

NF0: El sistema no tendrá administración para el manejo de retenciones.

NF1: El sistema no realizará cálculos de depreciaciones.

NF2: El sistema no tendrá administración para el manejo independiente de impuestos de ningún tipo.

NF3: El sistema no tendrá un plan de cuentas diferente para todos los usuarios, manejará un plan específico.

2.2. HERRAMIENTAS DEL SISTEMA

Para el desarrollo de esta disertación se eligió el lenguaje de programación PHP versión 5.3 debido a que es un lenguaje desarrollado específicamente para el ambiente web, además de todas las ventajas que se mostraron en el Capítulo 1. Como complemento a este lenguaje de programación se encuentra ajax, javascript, css.

Entre otras herramientas que se van a presentar se encuentra MySQL en su versión 5.0 como Motor de Base de Datos. El rendimiento que se tiene con este motor de bases de datos y PHP, es el necesario para poder realizar esta disertación, sus ventajas están presentadas en el Capítulo 1.

Para la creación de Diagramas UML se seleccionó Power Designer en su versión 12, ya que permite realizar las diagramaciones de una manera clara y entendible.

2.3. UML y Diagramas

El Lenguaje de Modelado Unificado es un lenguaje para el modelo de sistemas de software que sirve para visualizar, especificar, construir y documentar un sistema. Con este lenguaje se puede describir un sistema, así como también describir métodos o procesos.

En la versión 2.0 del Lenguaje UML, existen 13 diferentes tipos de diagramas. Este tipo de lenguaje se ajusta a cualquier tipo de metodología de desarrollo de software. Como en la presente disertación se maneja una metodología Extreme Programming, se debe realizar sólo los diagramas de Casos de Uso y el Diagrama de Clases.

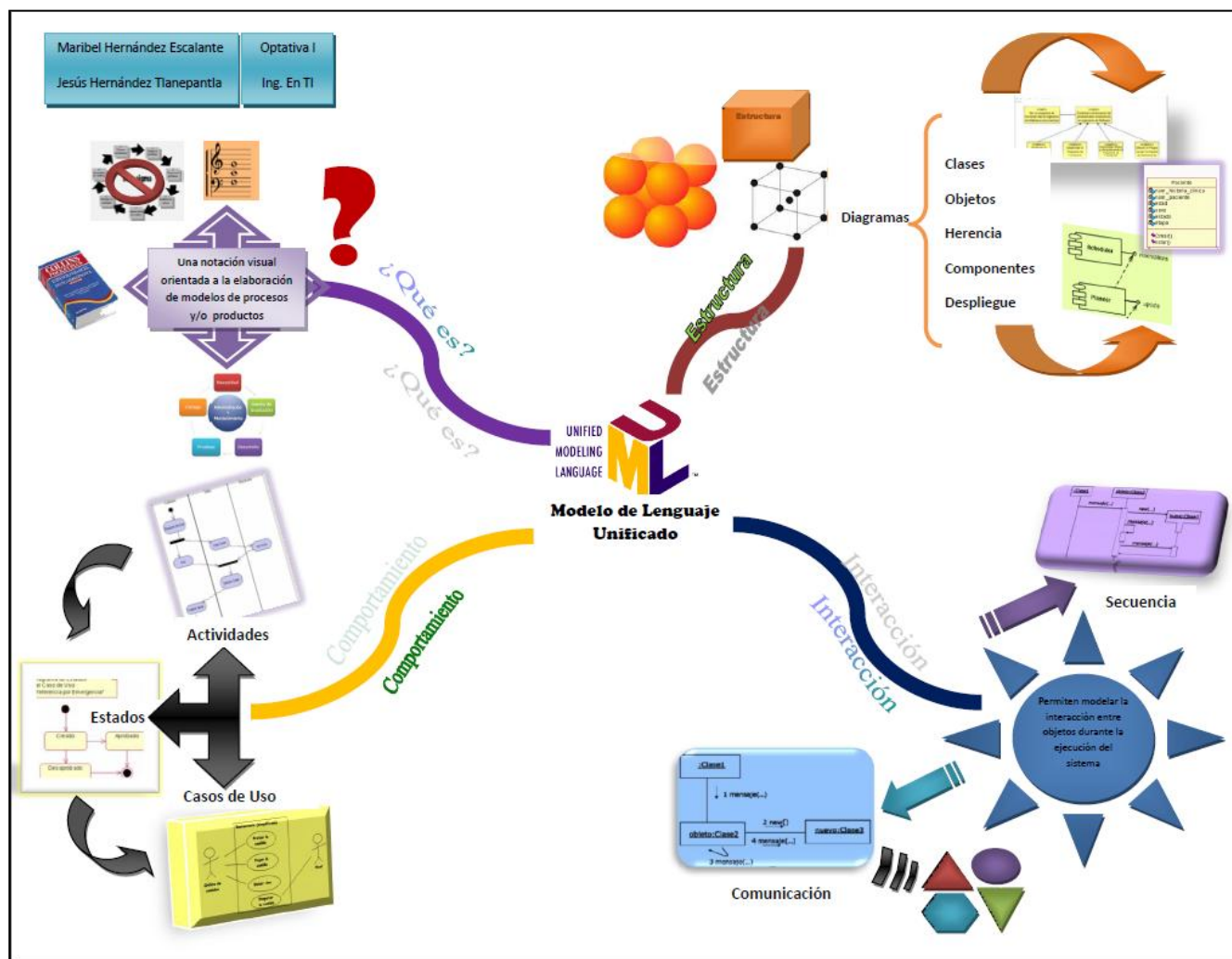


FIG 2-01 Mapa Mental UML [9]

2.3.1 DIAGRAMAS DE CASOS DE USO

2.3.1.1 DIAGRAMA GENERAL (NIVEL 0)

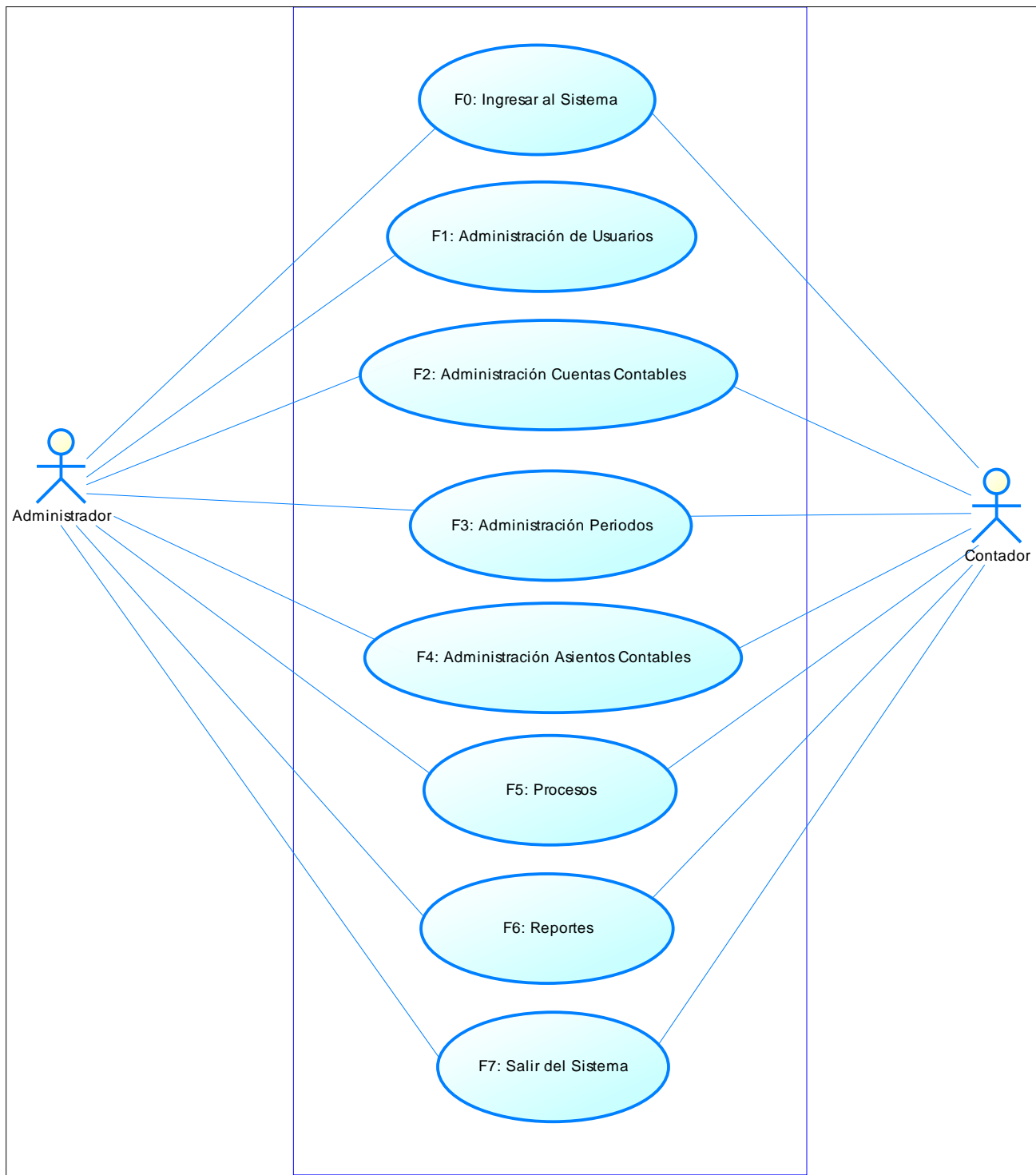


FIG 2-02 Diagrama de Casos de Uso General [A]

2.3.1.2 DIAGRAMA A DETALLE

F1: Administración para el manejo de Usuarios

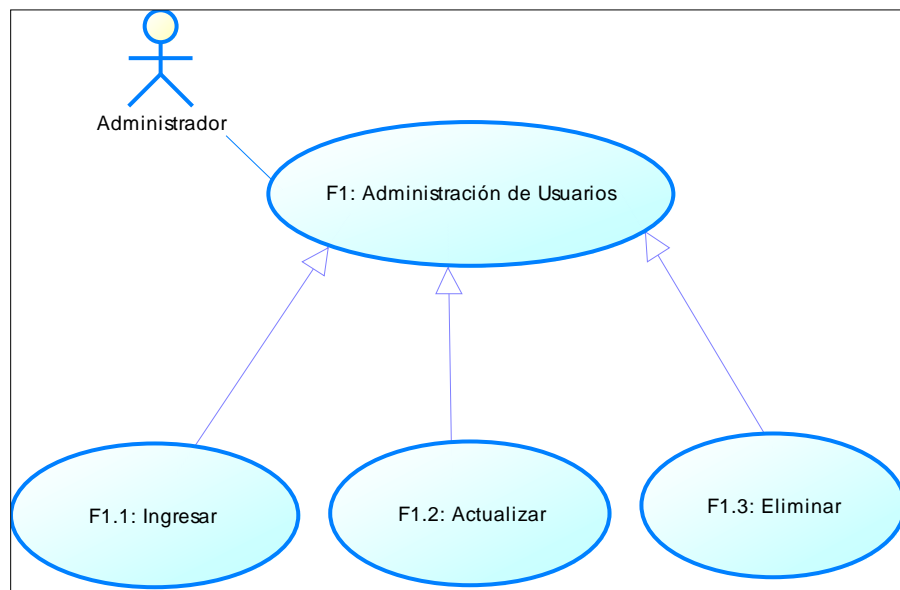


FIG 2-03 Diagrama de Casos de Uso – Administración de Usuarios [A]

F1.1: Ingreso de Usuarios

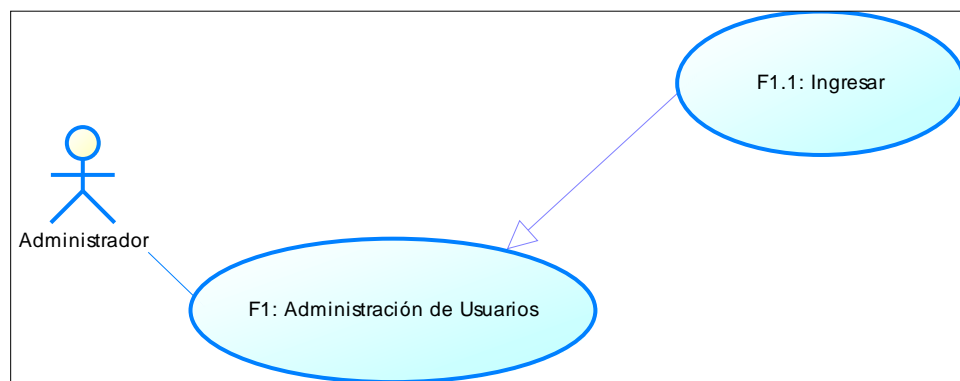


FIG 2-04 Diagrama de Casos de Uso – Ingreso de Usuarios [A]

F1.2: Actualización de Usuarios

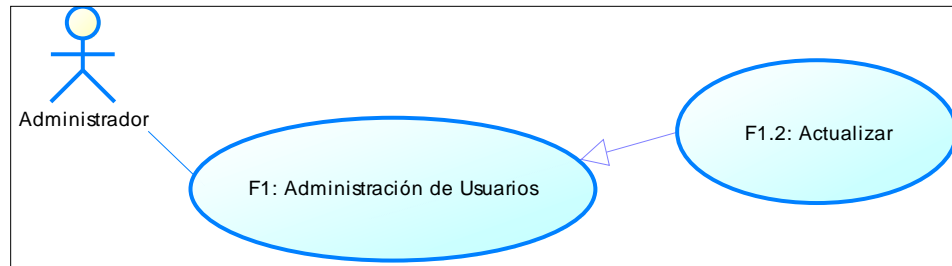


FIG 2-05 Diagrama de Casos de Uso – Actualización de Usuarios [A]

F1.3: Eliminación de Usuarios

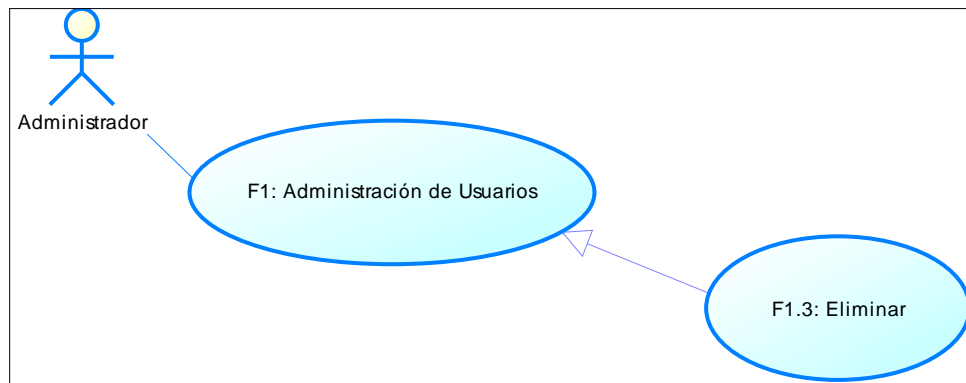


FIG 2-06 Diagrama de Casos de Uso – Eliminación de Usuarios [A]

PARA MÁS REFERENCIAS VER ANEXOS

2.3.2 DIAGRAMA DE CLASES

2.3.2.1 DIAGRAMA GENERAL

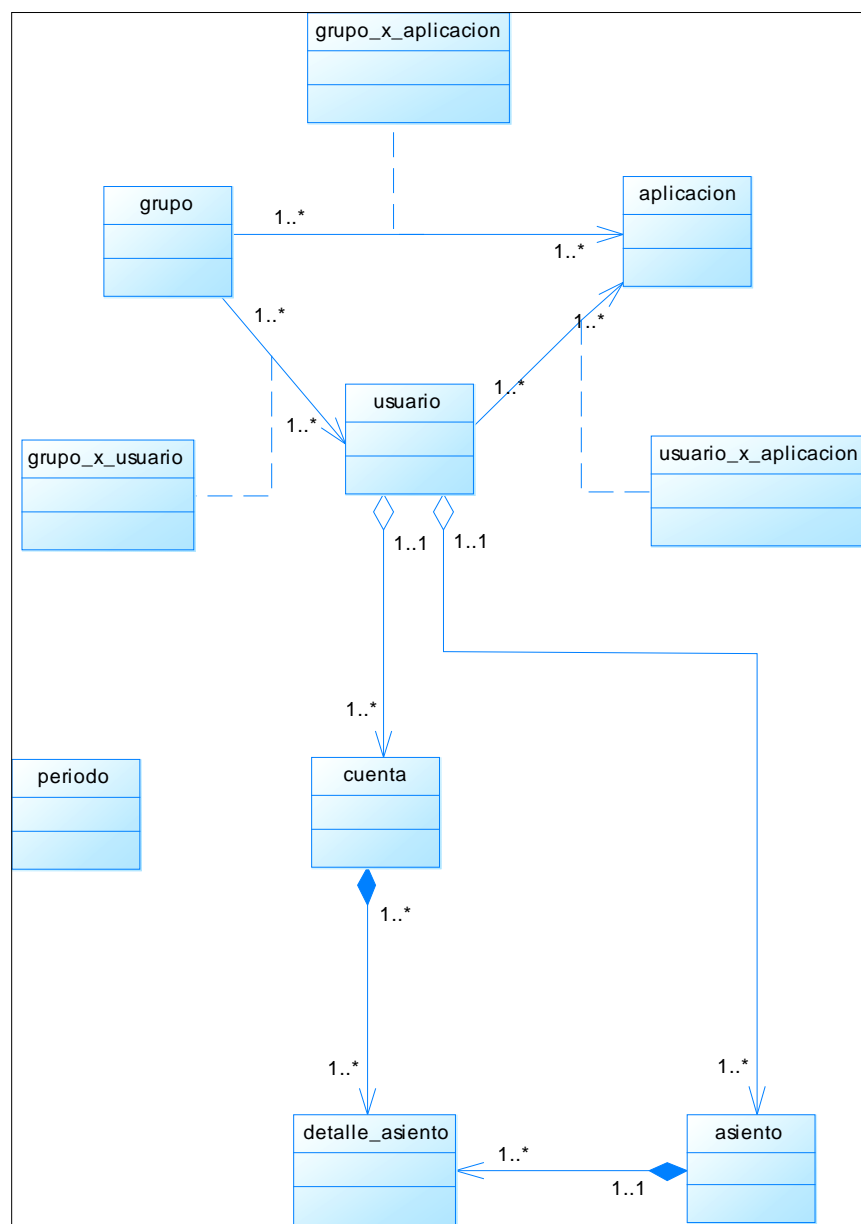


FIG 2-07 Diagrama de Clases General [A]

2.4. MODELO ENTIDAD – RELACIÓN

2.4.1 MODELO CONCEPTUAL

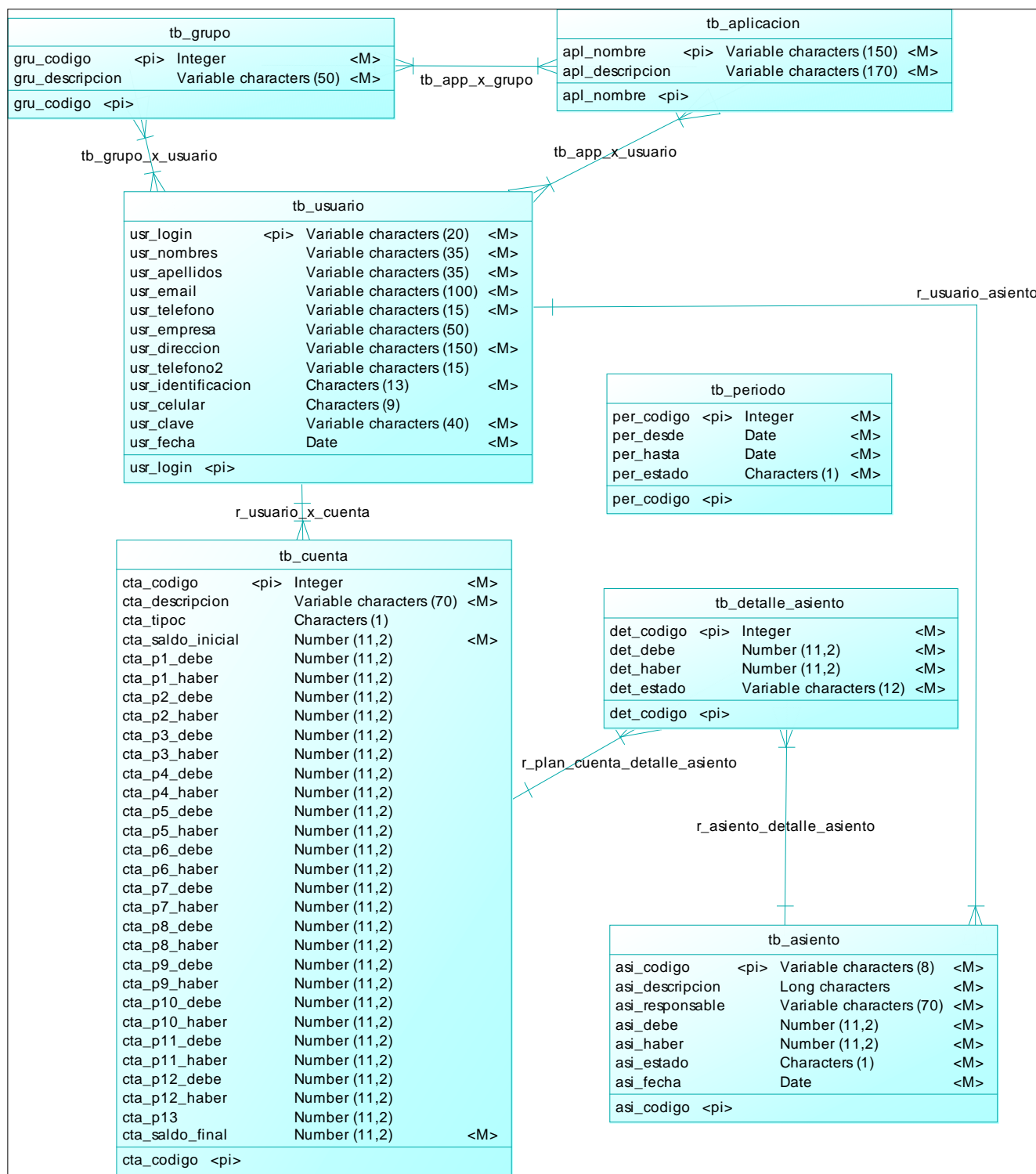


FIG 2-08 Modelo Conceptual [A]

2.4.2 MODELO FÍSICO

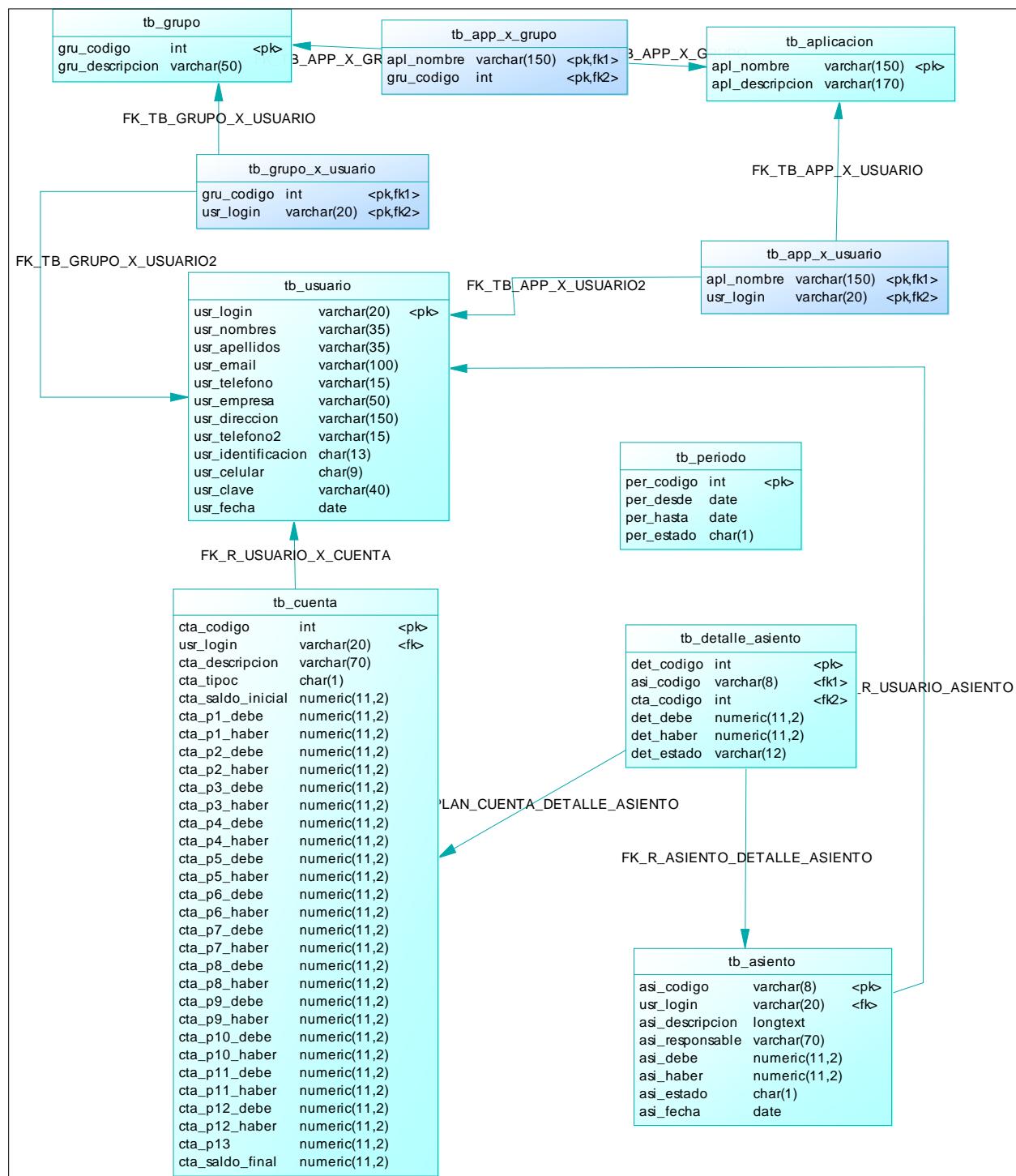
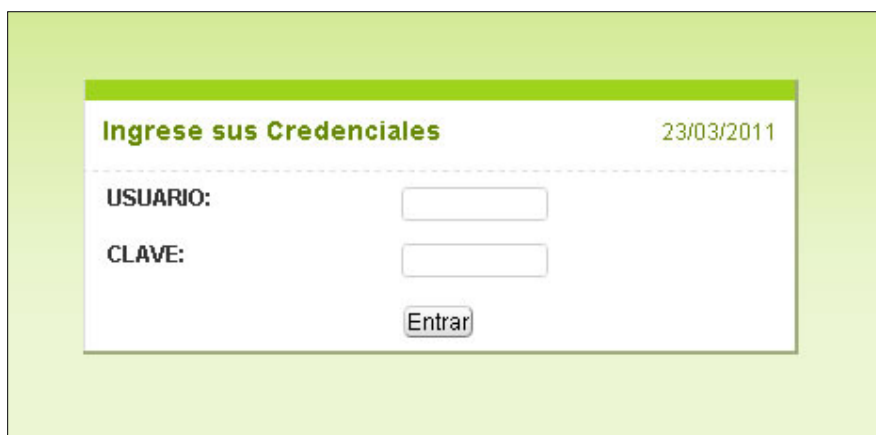


FIG 2-09 Modelo Físico [A]

2.5. PROTOTIPO DE INTERFAZ

2.5.1 LOGIN DE USUARIOS



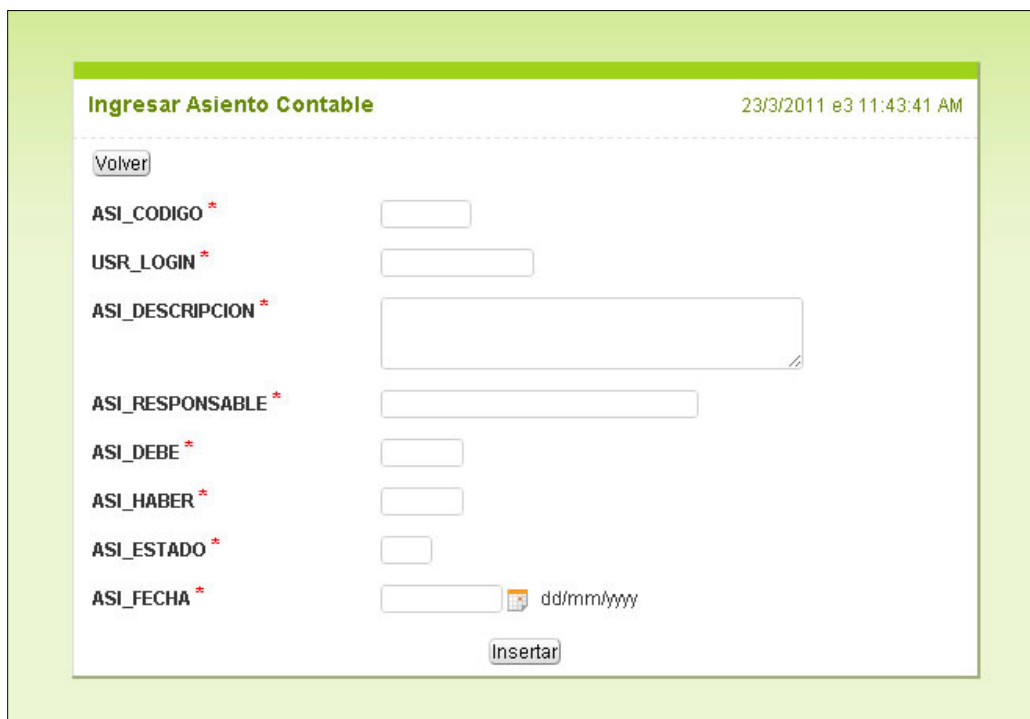
Ingrese sus Credenciales 23/03/2011

USUARIO:

CLAVE:

FIG 2-10 Ingreso de Usuarios [A]

2.5.2 PROTOTIPO DE INTERFAZ



Ingresar Asiento Contable 23/3/2011 e3 11:43:41 AM

ASI_CODIGO *

USR_LOGIN *

ASI_DESCRIPCION *

ASI_RESPONSABLE *

ASI_DEBE *

ASI_HABER *

ASI_ESTADO *

ASI_FECHA * dd/mm/yyyy

FIG 2-11 Interfaz Estándar [A]

2.5.3 PROTOTIPO DE ERROR



FIG 2-12 Prototipo de visualización de errores [A]

2.5.4 PROTOTIPO DE ERROR DE CAMPOS

A screenshot of a web application form titled "Ingresar Asiento Contable" with a timestamp "23/3/2011 e3 11:48:59 AM". The form contains eight input fields, each with a red asterisk indicating a required field. Below each input field is a red box with the text "Datos requeridos". The fields are: ASI_CODIGO, USR_LOGIN, ASI_DESCRIPCION, ASI_RESPONSABLE, ASI_DEBE, ASI_HABER, ASI_ESTADO, and ASI_FECHA. The ASI_FECHA field has a date picker icon and the format "dd/mm/yyyy". At the bottom of the form are two buttons: "Insertar" and "Cancelar".

FIG 2-13 Prototipo de visualización de errores por campo [A]

2.5.5 PROTOTIPO DE REPORTE

BALANCE CONTABLE AL 22 DE MARZO DE 2011 14:17					
CODIGO	CUENTA CONTABLE		SALDO INICIAL		SALDO FINAL
1	ACTIVOS				
1.1	ACTIVOS CORRIENTES				
1.1.1	BANCOS		100		
1.1.1.1	PICHINCHA		50		
1.1.1.2	PACIFICO		200		
1.1.1.3	PRODUBANCO		600		
2	PASIVOS				
2.1	PASIVOS CORRIENTES				
2.1.1	CXP				350
2.1.2	PROVEEDORES				600
		TOTAL:	950		950

FIG 2-14 Prototipo de visualización de reportes [A]

CAPÍTULO 3: IMPLEMENTACION

3.1. DESARROLLO Y OBSERVACIONES V1.0

3.1.1 DESARROLLO V1.0

A continuación se presenta el detalle del desarrollo del sistema en su versión 1.0:

- Ingreso de Usuario y Contraseña, con validaciones en contra de errores en cada uno de los campos utilizados.
- Registro de nuevos usuarios, para que puedan hacer uso del sistema.
- Menú de opciones para acceder a cada una de las opciones del sistema.
- Administración de Periodos en la cual se incluye:
 - Grid de visualización de registros, con sus opciones respectivas.
 - Formulario de ingreso, actualización y eliminación de periodos contables, con las validaciones necesarias a prueba de usuario.
- Administración de plan de cuentas en el cual se incluye:
 - Grid de visualización de registros, con sus opciones respectivas.
 - Formulario de ingresos, actualización y eliminación de cuentas contables.
- Administración de asientos contables:
 - Grid de visualización de asientos contables en base al periodo que se encuentra en estado activo.
 - Formulario de ingreso, actualización y eliminación de asientos contables.
 - Enlace a la administración de detalle de asientos.
- Administración de detalle de asientos, en la cual se define:
 - Grid de visualización del detalle perteneciente al asiento contable. Este grid a su vez, permite realizar el ingreso, actualización y eliminación de la información.

3.1.2 OBSERVACIONES DEL CLIENTE DE PRUEBA V1.0

Una vez presentada la versión 1.0, el cliente de prueba encontró las siguientes observaciones:

- Una vez que el usuario se registra, se debe cerrar la ventana de registro y redirigirse a la ventana de ingreso a la aplicación.

- Cuando el usuario se registra, debe existir una opción que permita ingresar el año fiscal.
- Una vez que se ha ingresado al sistema, se debe visualizar una imagen referente a contabilidad.
- En el menú de opciones:
 - Crear una opción que diga 'Generar plan Básico', el cual creará automáticamente un plan de cuentas por defecto.
 - Cambiar el nombre de la opción de 'estados contables', a 'periodos'.
 - Cambiar el nombre de la opción de 'cierres de mes', a 'cierres de periodos'
- En la administración del plan de cuentas:
 - En el grid de visualización de datos:
 - No debe mostrarse la columna usuario.
 - Se debe renombrar el botón de 'alterar' por 'número de registros'.
 - En el formulario de ingreso:
 - El tipo de cuenta debe presentarse como maestro o detalle.
 - El saldo inicial debe permitir ingresar valores numéricos negativos
 - Cuando se haya ingresado exitosamente la información debe redirigirse al grid de visualización del plan de cuentas.
- En la administración de periodos contables:
 - En el formulario de ingreso:
 - Siempre que se crea un nuevo periodo tiene un estado de inactivo.
 - Una vez que se ingresa un nuevo periodo, se debe redirigir al grid de visualización de registros.
- En la administración de asientos contables:
 - En el grid de visualización de datos:
 - No debe mostrarse la columna usuario.
 - Deben mostrarse los registros ordenados por el periodo activo.
 - En el formulario de ingreso:
 - La fecha del asiento tiene que ser por defecto la del sistema.
 - El valor del campo debe y el valor del campo haber, tienen que ser solo de lectura.
 - El estado de los asientos nuevos debe ser abierto y no debe visualizarse.

- Si se elimina el asiento deben eliminarse los detalles asignados a este y redirigirse al grid de visualización de registros.
 - No se puede ingresar asientos en periodos cerrados.
- En la administración de detalle de asientos:
 - El estado del detalle del asiento siempre tiene que ser abierto.
 - Si no cuadra el valor del debe, con el valor del haber, debe mostrarse un mensaje de advertencia.
- Proceso de Mayorización
 - Sólo se mayorizará los asientos contables que estén cuadrados y que estén con estado de abierto.
 - Se registra el valor del debe y del haber en el plan de cuentas, correspondiente al periodo activo.
 - Una vez que todos los detalles estén mayorizados, cambia el estado del asiento a mayorizado.
 - Solo se podrán actualizar los asientos con estado abierto.
- Proceso de Cierre de Mes
 - Todos los asientos contables deben estar mayorizados.
 - Se debe actualizar el estado del periodo actual a cerrado y se activa el siguiente periodo.
 - Se debe sumar los valores totales del debe y del haber de las cuentas del detalle y colocarlas en la cuenta maestra.
- Proceso de Cierre Anual
 - El sistema debe validar que todos los asientos estén mayorizados, y que todos los periodos contables estén cerrados.
 - Se debe actualizar el valor del saldo final de cada una de las cuentas.
 - Debe mostrarse una advertencia que diga: '¿Seguro que quiere ejecutar el proceso de cierre de año?'
 - Se debe poner en 0, todos los valores de los periodos del plan de cuentas.
 - Debe actualizarse el año fiscal, por el siguiente año.
 - Se debe eliminar todos los asientos y detalles del año fiscal cerrado.
 - El saldo inicial de todas las cuentas debe ser 0.

3.1.3 DESARROLLO V1.5

Después de las observaciones dadas por el usuario de prueba, se realizó la versión 1.5 del sistema, en donde se incluyen todas las observaciones dadas por el usuario. Además de los cambios pedidos, también se generaron validaciones para cada uno de los procesos del sistema.

3.1.4 OBSERVACIONES DEL CLIENTE DE PRUEBA V1.5

Se encontraron las siguientes observaciones por parte del usuario de prueba al presentarle la versión 1.0:

- No debe existir la opción de generar plan de cuentas, ni generar periodos contables , estos deben generarse al momento de que el usuario ingresa a la aplicación.
- Debe poder ejecutarse el proceso de cierre de mes, aunque no se hayan registrado asientos contables para el período abierto.
- Al ejecutarse el cierre de mes, deben sumarse los totales de las cuentas de detalle a las cuentas padre, con un máximo de tres niveles.
- En los procesos de mayorización, de cierre de periodo, y de cierre anual, debe presentarse un mensaje de confirmación de ejecución del proceso.
- Deben generarse los reportes de cierre de periodo y de cierre anual.

3.1.5 DESARROLLO V2.0

Para finalizar con el proyecto de disertación de tesis, se realizaron los cambios pedidos por el usuario de prueba en el punto anterior. Con esto el desarrollo termina satisfactoriamente para el usuario final.

CAPÍTULO 4: PRUEBAS Y SERVICIO

4.1. PRUEBAS

4.1.1 PRUEBAS UNITARIAS

A medida que se ha ido desarrollado el sistema se han ejecutado diferentes tipos de pruebas para verificar la funcionalidad del software e ir corrigiendo los errores presentados. Estas pruebas se han realizado en cada una de las administraciones presentadas en este documento, así como en los procesos especificados anteriormente.

- Las pruebas unitarias realizadas son las siguientes:
- Pruebas de validaciones de campos.
- Prueba de visualización de registros.
- Prueba de ingreso, actualización y eliminación de información.
- Ejecución de procesos: mayorización, cierre mensual, cierre anual.

Cabe resaltar que las pruebas unitarias de este sistema fueron desarrolladas por el usuario de prueba que a su vez brindó las observaciones necesarias para un correcto funcionamiento de la aplicación.

4.1.2 PRUEBAS CONJUNTAS

Una vez finalizadas las pruebas unitarias se da paso a las pruebas conjuntas. Estas pruebas ayudan a verificar problemas en multiaccesos o cruce de la información, además de validar el funcionamiento correcto del sistema. Los errores encontrados, los cuales fueron mínimos, fueron corregidos satisfactoriamente.

Las pruebas conjuntas se las ejecutaron con varias personas utilizando el sistema al mismo tiempo.

Las pruebas en conjunto mostraron que:

- Todos los campos están validados correctamente.
- Todas las administraciones funcionan según los requerimientos presentados.

- Todos los procesos se ejecutan satisfactoriamente y muestran los resultados correctos.
- Los balances reflejan información correcta.

4.1.3 PRUEBAS CONCURRENTES

Para finalizar el procesos de pruebas para el control de errores del software desarrollado, se debe ejecutar un número de pruebas concurrentes. Estas pruebas han sido realizadas de la siguiente manera:

- Tres usuarios se han registrado dentro de la aplicación.
- Se generó el plan de cuentas para cada uno y los periodos con todo éxito.
- Se generaron asientos contables, con sus detalles respectivos, sin ningún problema.
- Los tres usuarios ejecutaron los procesos de mayorización, de cierre de periodo y de cierre anual al mismo tiempo. Los resultados fueron:
 - No hubo problema en ninguno de los tres usuarios.
 - La base de datos manejo cada uno de los procesos como si estuvieran en cola.
 - No se encontraron errores, ni al momento de la ejecución de los procesos, ni al momento en que se realizaron las actualizaciones en aplicación generadas por cada uno de los procesos.

4.2. SERVICIO

4.2.1 SERVICIO EN VIVO

Finalizadas las pruebas, se debe ‘implantar’ el software como un servicio en vivo, esto quiere decir, que se deben mover los archivos de un ambiente local de desarrollo a un ambiente en externo de producción. Para esto se copian todos los archivos a un hosting específico que nos brinde salida a internet.

Con los archivos ya en el hosting, procedemos a utilizar el sistema a través de internet y hacemos las pruebas correspondientes para verificar que todo funcione correctamente.

En este caso no se encontraron errores al momento de tener el sistema en internet. Para tener más información de los requisitos básicos que se necesitan y saber cómo funciona el sistema,

es recomendable leer el manual técnico y el manual de usuario que se encuentran en los anexos.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

5.1.1 CONCLUSIONES DEL CAPÍTULO I

- El Internet es una red de computadores que ha permitido unir las fronteras a nivel mundial, logrando que personas en diferentes ciudades o países puedan comunicarse.
- Hypertext Pre Processor es el lenguaje de programación más utilizado en la actualidad para el diseño de sistemas en plataforma web y cada día que avanza, miles de usuarios están empezando a adoptar éste lenguaje como estándar en el desarrollo de nuevas aplicaciones.
- El paradigma Cloud Computing constituye una realidad para muchas empresas y a través de cada una de sus capas; Saas, IaaS, PaaS está brindando un nuevo tipo de servicio que de momento en el Ecuador no se conoce en gran magnitud.
- La metodología Extreme Programming ayuda al diseño y desarrollo de sistemas en plataforma web siempre y cuando se trabaje de la mano con el cliente y se entreguen avances periódicos. Si no se trabaja conjuntamente con el cliente, es necesario utilizar otro tipo de metodologías que permita un levantamiento de requerimientos cien por ciento fiable.
- La Contabilidad permite medir y conocer la situación financiera de una empresa, para en base a indicadores, poder tomar decisiones que puedan sustentar el progreso de la organización.

5.1.2 CONCLUSIONES DEL CAPÍTULO II

- La identificación de requerimientos permite determinar hasta dónde quiere llegar el cliente y cuáles son sus necesidades primarias y secundarias. Con el levantamiento de requerimientos se puede tener una idea clara de lo que va a ser la aplicación final.
- Los requerimientos no funcionales especifican los procesos que no estarán presentes en el desarrollo de la aplicación. Sin embargo, se los puede tomar en cuenta para una futura implementación.
- Los diagramas UML permiten construir, documentar y visualizar como va a estar desarrollada una aplicación. El Lenguaje UML es independiente de cualquier arquitectura de programación y en la actualidad está siendo reemplazado por nuevas técnicas de construcción de aplicaciones.
- El buen desarrollo del modelo Entidad – Relación de la Base de Datos de la aplicación en plataforma web permitirá un mejor rendimiento y funcionamiento al momento de la usabilidad del mismo.
- Determinar cuáles son las mejores herramientas para el desarrollo de una aplicación en plataforma web, es fundamental para lograr un buen rendimiento y poder cumplir con los requerimientos especificados por un cliente.

5.1.3 CONCLUSIONES DEL SISTEMA DESARROLLADO

- El sistema fue desarrollado con una metodología extrema que funcionó a la perfección con el usuario final, logrando los resultados esperados en el levantamiento de requerimientos.
- Con las pruebas realizadas, se lograron identificar algunas falencias, las cuales fueron corregidas inmediatamente para no perjudicar el proceso de desarrollo.
- Al desarrollar una aplicación con un modelo de distribución como es SaaS, el usuario accede automáticamente a tener muchas ventajas, como lo son, el espacio en disco, la

información en tiempo real, no tiene que preocuparse de las seguridades, entre otras cosas.

5.1.4 CONCLUSIONES QUE VALIDEN LA HIPÓTESIS

- La aplicación está desarrollada en plataforma web, con herramientas libres y está alojada en un servidor compartido externo al cual se puede acceder a través del internet.
- Se utiliza el concepto de Saas (Software as a Service), por lo que cada usuario puede entrar a la aplicación desde cualquier computador al servidor que contenga la aplicación, por lo que el usuario no debe preocuparse más que de la funcionalidad del mismo. Saas es realizar software como un servicio en internet, que beneficia a las empresas brindándoles seguridad, confianza y la información en cualquier lugar del mundo y en tiempo real.
- El proveedor del servicio se encarga del mantenimiento de servidor, del funcionamiento correcto de la aplicación, del espacio en disco que tenga el servidor, entre otras cosas, haciendo que el usuario que va a obtener el servicio se sienta seguro y no deba preocuparse de estos detalles.
- En la actualidad y en el futuro SAAS será la solución del futuro, porque garantiza a las empresas tener su información disponible las 24 horas del día, los siete días de la semana. Además de esto permite abaratar costos y generar mayor beneficio.

5.1.5 RECOMENDACIONES PARA EL USO DEL SISTEMA

- Es importante leer la guía de usuario antes de utilizar la aplicación, para un buen uso de la misma.
- Tener una buena conexión de Internet permitirá mayor rapidez en las transacciones o solicitudes que se realicen dentro de la aplicación.

- Si existe dudas al respecto sobre el sistema, se debe preguntar directamente al administrador vía correo electrónico.
- Se debe conocer las limitaciones que tiene el sistema y cuál es el alcance del mismo para evitar problemas de funcionalidad o de necesidad por parte del usuario.

5.1.6 RECOMENDACIONES PARA EL LEVANTAMIENTO DE REQUERIMIENTOS

- Abarcar varios aspectos relacionados a lo que el cliente pueda necesitar, permite englobar de mejor manera lo que será el funcionamiento del sistema.
- No quedarse con ninguna duda sobre algún requerimiento, ya que esta duda podría causar un mal diseño e implementación al momento del desarrollo.
- Nunca decir que algo no se puede hacer, sin embargo, se debe explicar el por qué sería mejor no hacerlo o el tiempo y el costo que abarcaría realizar un requerimiento.
- Tener presente cada uno de los requerimientos funcionales y no funcionales, ayuda al proceso de diseño, desarrollo e implementación del sistema.

5.1.7 RECOMENDACIONES TÉCNICAS

- Leer el manual técnico puede brindar un mayor conocimiento sobre cómo está desarrollada la aplicación y que herramientas se están utilizando.
- Conocer sobre el lenguaje de programación PHP y como trabaja directamente con MySQL.
- Es importante comprender el modelo de base de datos y donde se almacena la información, para poder realizar ajustes o agregar nuevas funcionalidades a la aplicación.
- Conocer sobre el lenguaje de programación Javascript para entender el funcionamiento de Ajax dentro de la aplicación.

- Acceder a la aplicación con un computador con procesador mínimo dualcore y 2Gb de memoria RAM para una navegabilidad rápida.

5.1.8 RECOMENDACIONES GENERALES

- Si se tiene un usuario y una contraseña de acceso, no dar las credenciales a ninguna persona, ya que puede tener problemas de seguridad y tal vez los datos podrían ser alterados.
- Cambiar periódicamente la contraseña de acceso además de su usuario.
- El sistema debería ser manejado por personas que tengan un conocimiento en el área contable, caso contrario podría haber inconvenientes en el manejo o funcionamiento del sistema y los reportes no serán los exactos.

5.1.9 RECOMENDACIONES PARA LA FACULTAD DE INGENIERIA

- Es importante que la Facultad de Ingeniería vaya de la mano con las nuevas tendencias tecnológicas para brindar un mayor nivel educacional a sus estudiantes.
- Como objetivo primordial la facultad tiene que plantearse el empezar a incorporar entre su malla curricular la programación orientada a la web como fundamentos principales, ya que es para allá a donde va el mundo de la programación.
- Conseguir gente especializada y preparada para los nuevos cambios tecnológicos, en redes de la comunicación, desarrollo de aplicaciones orientadas a SaaS, manejo y generación de nuevos proyectos, entre otras cosas.

BIBLIOGRAFÍA

- [A]. CARRASCO VÁSQUEZ, E. R. (14 de Febrero de 2011). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA CONTABLE BÁSICO PARA USO LIBRE EN PLATAFORMA WEB. Quito, Pichincha, Ecuador.

- [1]. Wikipedia. (25 de 01 de 2011). Software Libre. Recuperado el 25 de Enero de 2011, de Wikipedia: http://es.wikipedia.org/wiki/Software_libre

- [2]. Uberbin. (28 de 03 de 2005). Ajax. Recuperado el 21 de Enero de 2011, de Uberbin: <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>

- [3]. Wikipedia. (14 de 02 de 2011). MySQL. Recuperado el 10 de Enero de 2011, de Wikipedia: <http://es.wikipedia.org/wiki/MySQL>

- [4]. Tecknohints. (10 de 12 de 2009). Cloud Computing. Recuperado el 01 de Febrero de 2011, de Tecknohints: <http://tecknohints.wordpress.com/2009/12/10/cloud-computing/>

- [5]. Contactdubai. (13 de 08 de 2009). Cloud Computing orientado al servicio. Recuperado el 01 de Febrero de 2011, de Contactdubai: <http://contactdubai.com/webhosting/advantages-of-cloud-computing>

- [6]. DELL. (17 de Enero de 2011). Evolución de la computación en las nubes. Recuperado el 17 de Enero de 2011, de DELL: http://www.perotsystems.com/MediaRoom/WhitePapers/Cloud_Computing

- [7]. Saasmania. (14 de Septiembre de 2008). Infraestructura como un servicio. Recuperado el 03 de Febrero de 2011, de Saasmania: <http://www.saasmania.com/2008/09/14/directorio-cloud-computing/>

- [8]. Letelier, Patricio y Penadés Carmen (Enero 2004). Metodologías ágiles para el desarrollo de software: extreme Programming PDF, páginas: 6-15.

- [9]. Mendoza, Angel (25 de Abril de 2011). Mapa mental del Lenguaje de Modelado Unificado, <http://aemr-dasl.blogspot.com/2010/07/mapa-mental-del-uml.html>.

OTRAS REFERENCIAS

http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema
<http://www.maestrosdelweb.com/editorial/cloud-computing-nueva-era-de-desarrollo/>
http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_nube
<http://soaagenda.com/journal/articulos/arquitectura-orientada-a-servicios/>
http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios
<http://www.iprofesional.com/notas/46399-Que-es-SOA-la-arquitectura-orientada-a-servicios.html>
<http://www.bsc.co.ve/index.php/soa>
http://www.programacion.com/articulo/por_que_elegir_php_143
<http://www.tech-faq.com/es/pros-de-php.html>
<http://www.lawebera.es/manuales/php/2-2.php>
<http://www.hackingballz.com/herramientas/manual-oficial-de-php/features.html>
<http://codigoprogramacion.com/programacionweb/49-comparacion-php-jsp-asp.html>
<http://es.wikipedia.org/wiki/AJAX>
<http://www.webexperto.com/articulos/art/223/que-es-ajax/>
<http://www.tufuncion.com/ventajas-ajax>
<http://www.ajaxya.com.ar/>
<http://tecenciente.osmosislatina.com/cursos/ajax/basico.htm>
<http://es.wikipedia.org/wiki/HTML>
<http://www.monografias.com/trabajos7/html/html.shtml>
<http://www.gnu.org/licenses/licenses.es.html>
<http://www.arues.com/beneficios.htm>
<http://es.wikipedia.org/wiki/PHP>
<http://www.desarrolloweb.com/articulos/392.php>
<http://www.maestrosdelweb.com/editorial/phpintro/>
http://es.wikipedia.org/wiki/HTML_5
<http://es.wikipedia.org/wiki/Contabilidad>
http://www.wikilearning.com/monografia/contabilidad_basica/5926
<http://www.temasdeclase.com/libros%20gratis/teoriabasica/capuno/portada.htm>

ANEXOS

1. DIAGRAMAS DE CASOS DE USO A DETALLE – OTRAS ADMINISTRACIONES Y PROCESOS DEL SISTEMA

F2: Administración para el manejo de Cuentas Contables

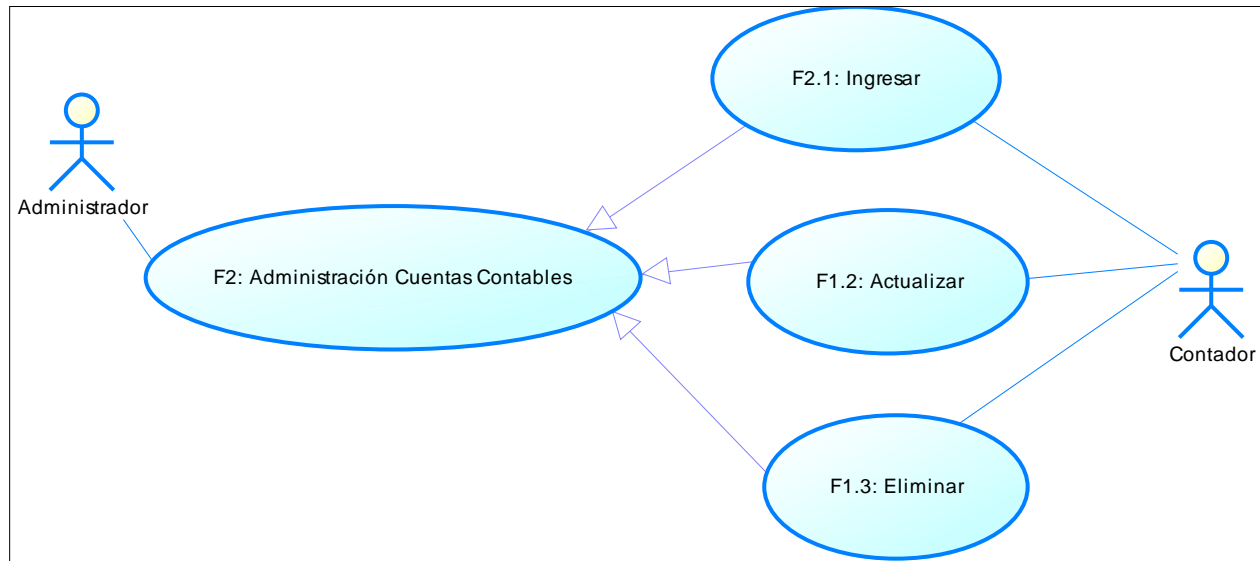


FIG Anexos-01 Diagrama de Casos de Uso – Administración Cuentas [A]

F2.1: Ingreso del de Cuentas Contables

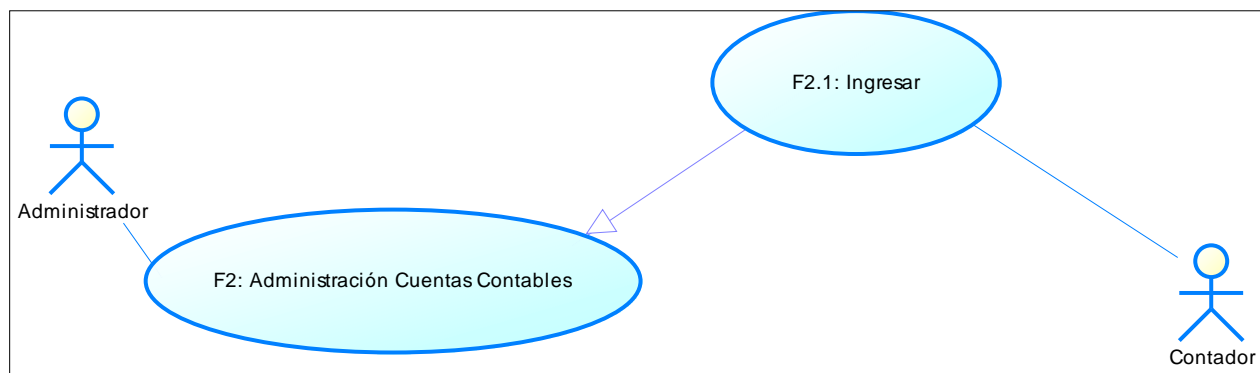


FIG Anexos-02 Diagrama de Casos de Uso – Ingresar Cuentas [A]

F2.2: Actualización de Cuentas Contables

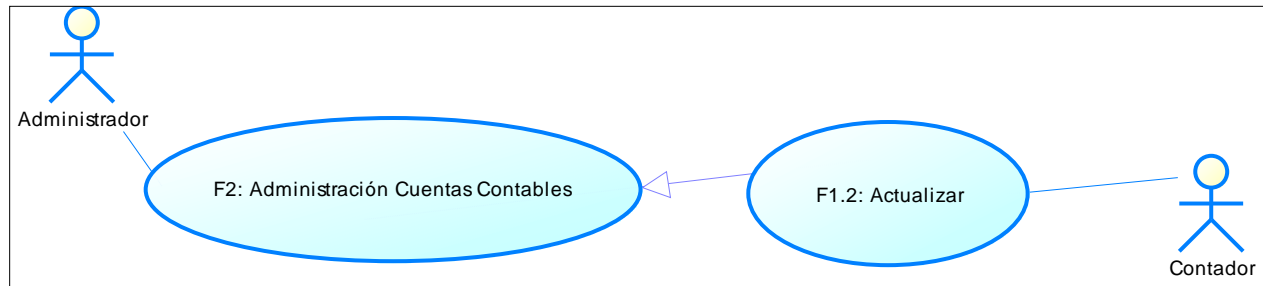


FIG Anexos-03 Diagrama de Casos de Uso – Actualizar Cuentas [A]

F2.3: Eliminación de Cuentas Contables

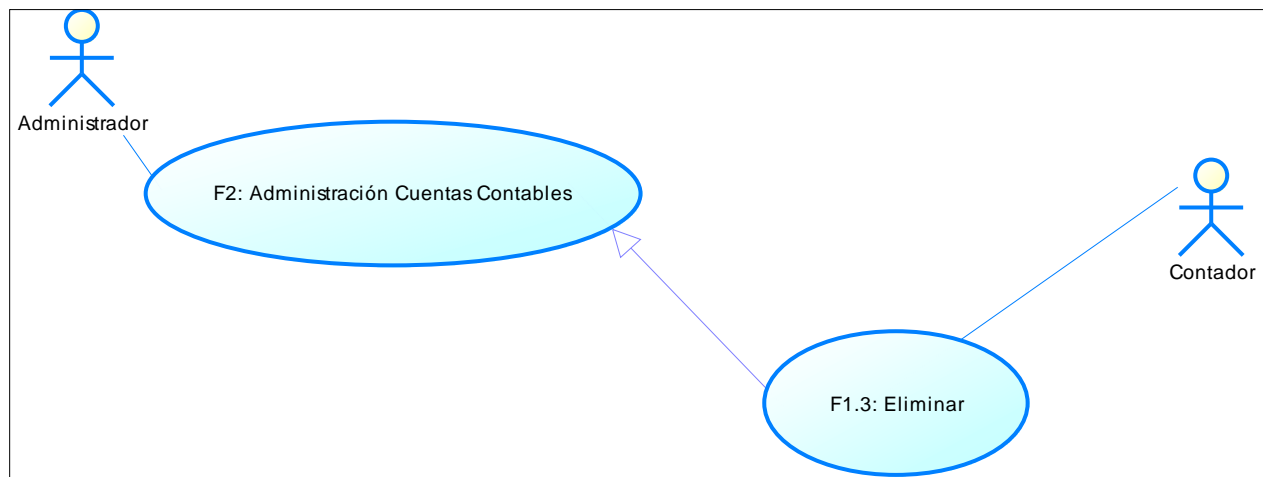


FIG Anexos-04 Diagrama de Casos de Uso – Eliminar Cuentas [A]

F3: Administración para el manejo de Periodos

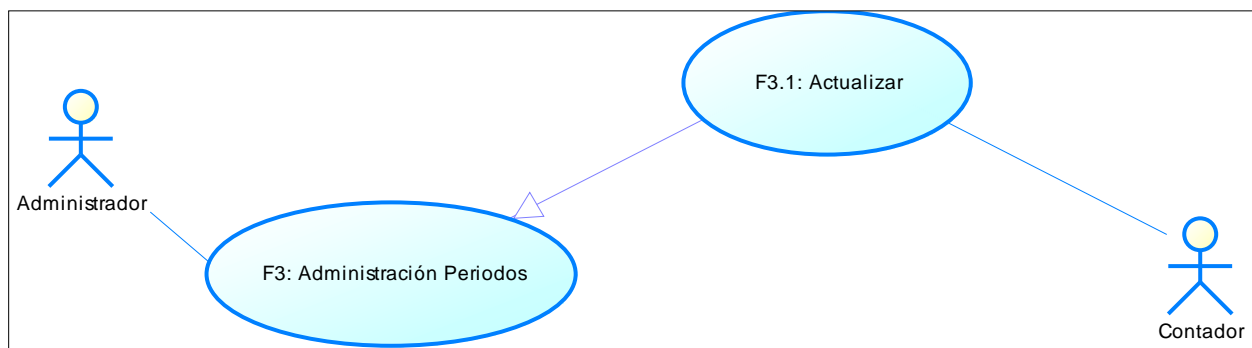


FIG Anexos-05 Diagrama de Casos de Uso – Administración de Periodos [A]

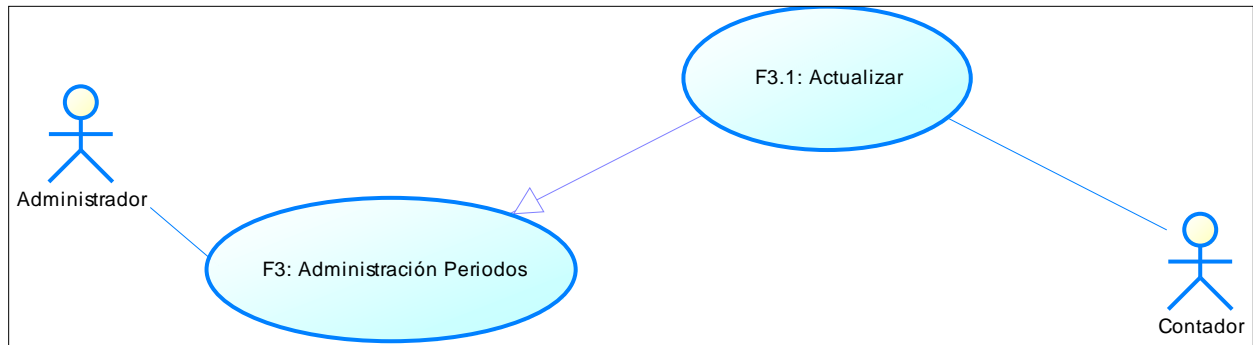
F3.1: Actualización de Periodos

FIG Anexos-06 Diagrama de Casos de Uso – Actualizar Periodos [A]

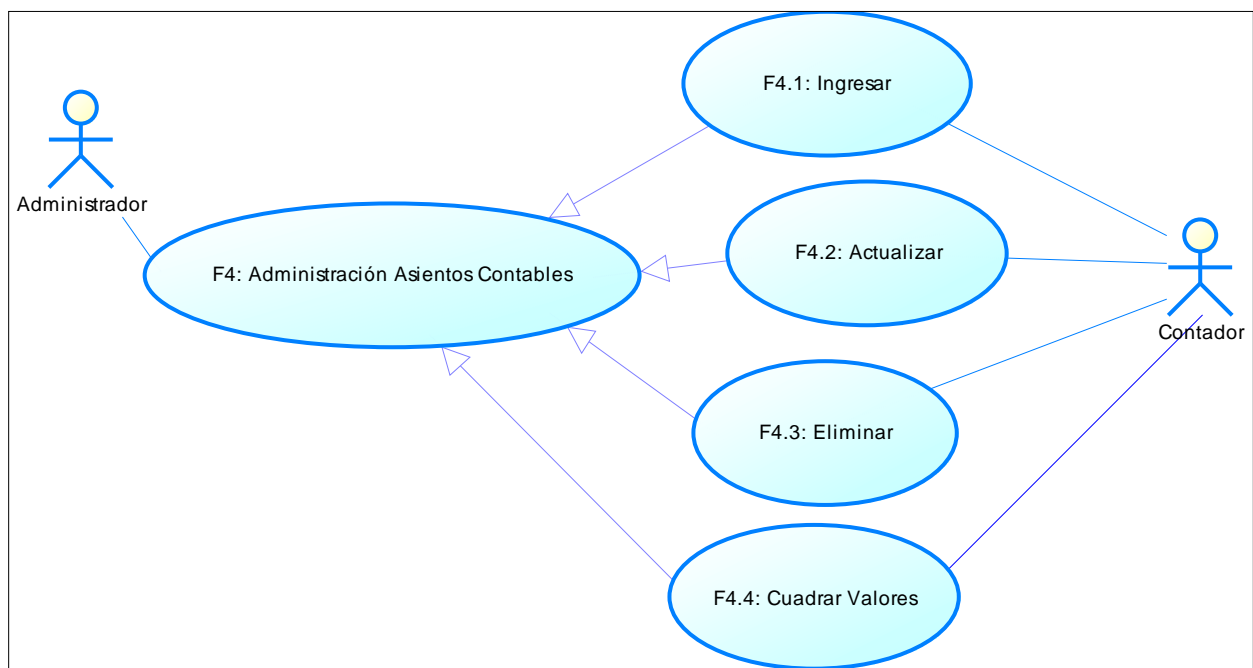
F4: Administración para el manejo de Asientos Contables

FIG Anexos-07 Diagrama de Casos de Uso – Administración de Asientos Contables [A]

F4.1: Ingreso de Asientos Contables

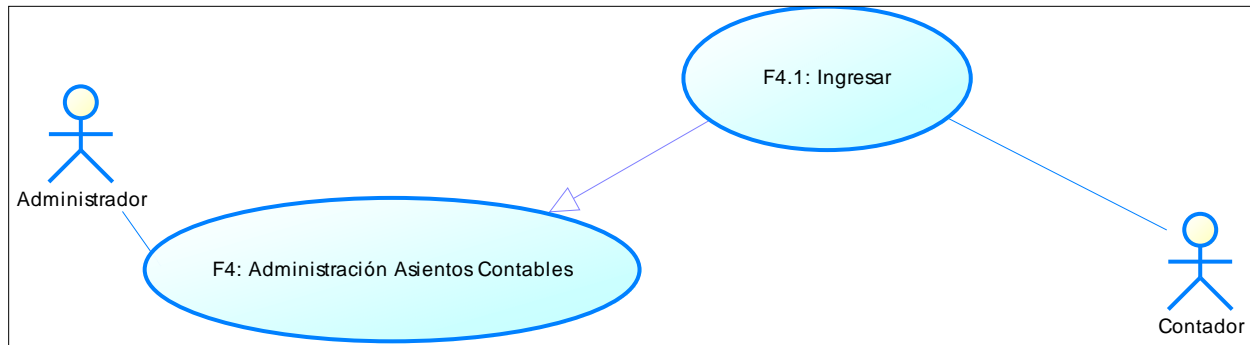


FIG Anexos-08 Diagrama de Casos de Uso – Ingresar Periodos [A]

F4.2: Actualización de Asientos Contables

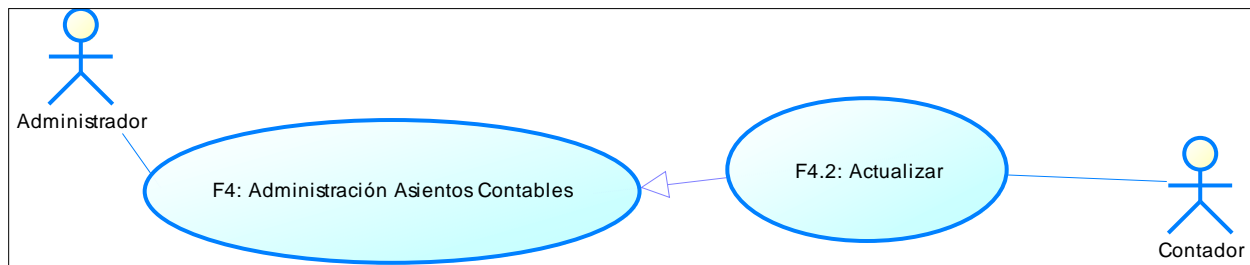


FIG Anexos-09 Diagrama de Casos de Uso – Actualizar Periodos [A]

F4.3: Eliminación de Asientos Contables

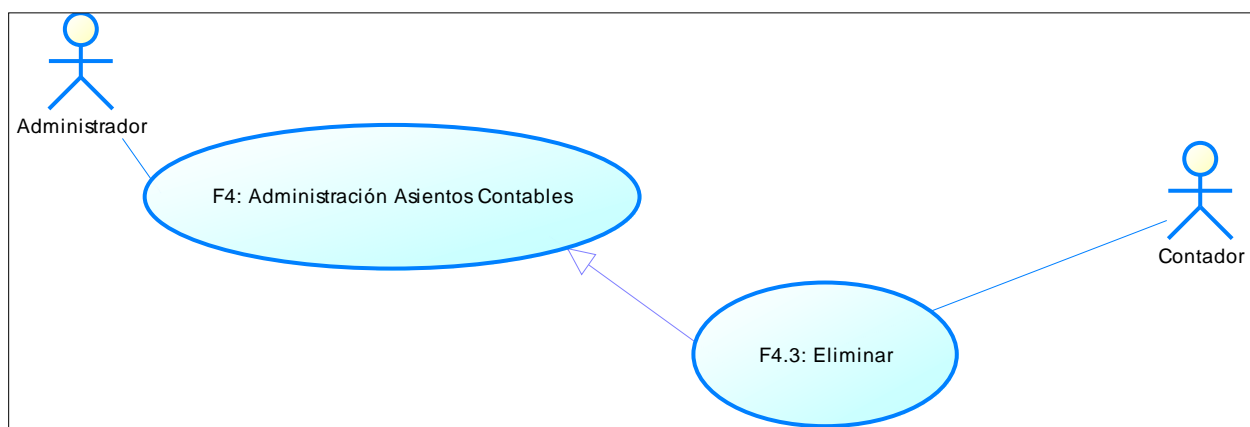


FIG Anexos-10 Diagrama de Casos de Uso – Eliminar Periodos [A]

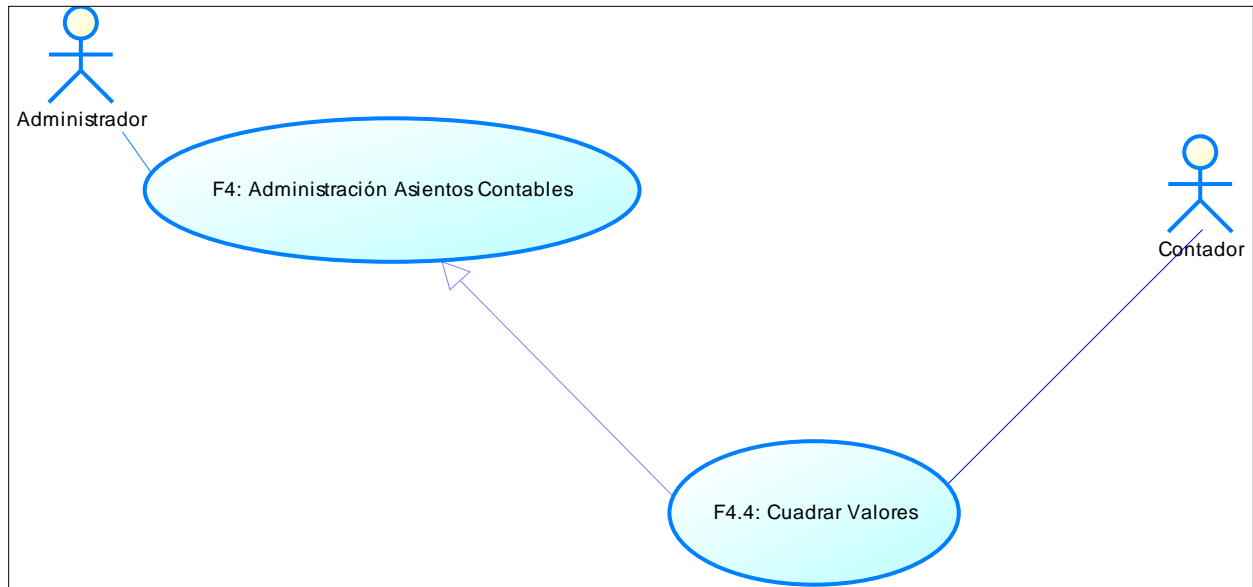
F4.4: Cuadre de Valores

FIG Anexos-11 Diagrama de Casos de Uso – Cuadrar Valores [A]

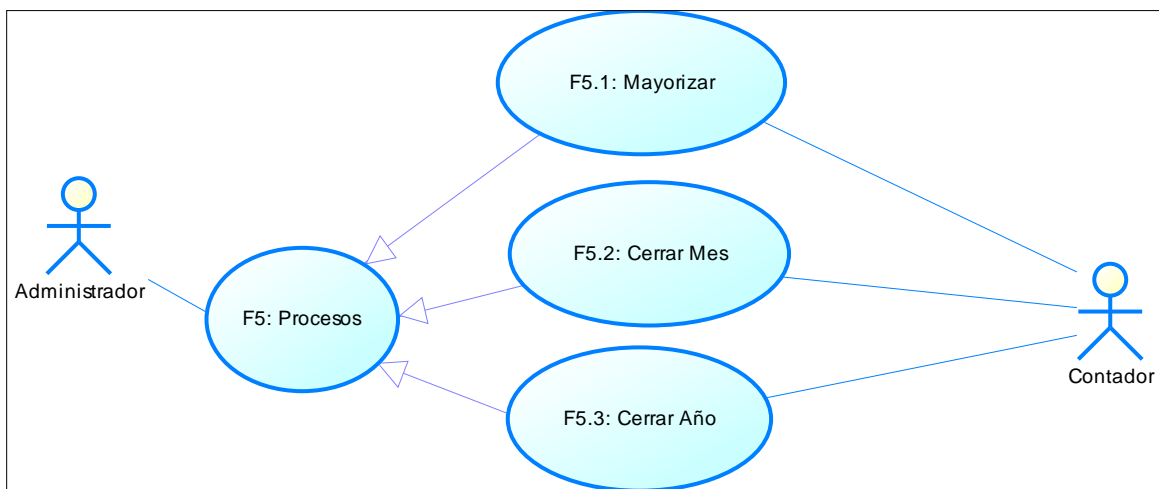
F5: Procesos prioritarios del Sistema Contable

FIG Anexos-12 Diagrama de Casos de Uso – Procesos del Sistema [A]

F5.1: Mayorización

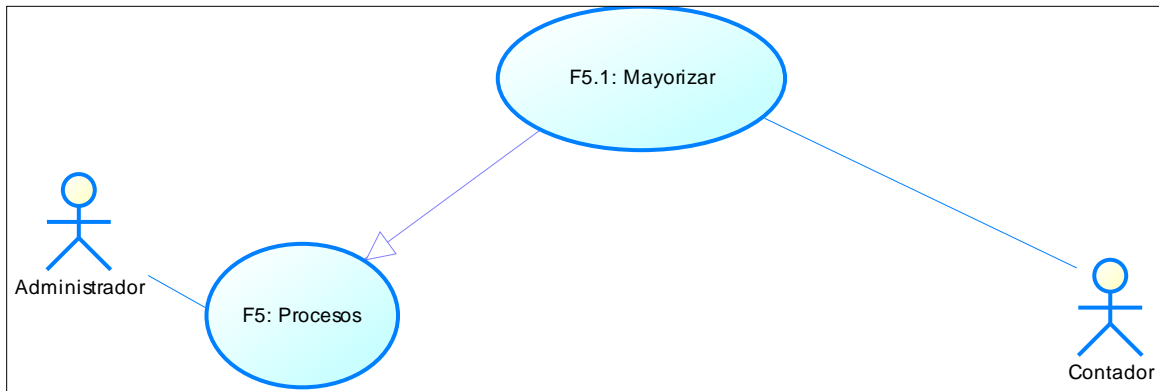


FIG Anexos-13 Diagrama de Casos de Uso – Mayorizar [A]

F5.2: Cierre de Mes

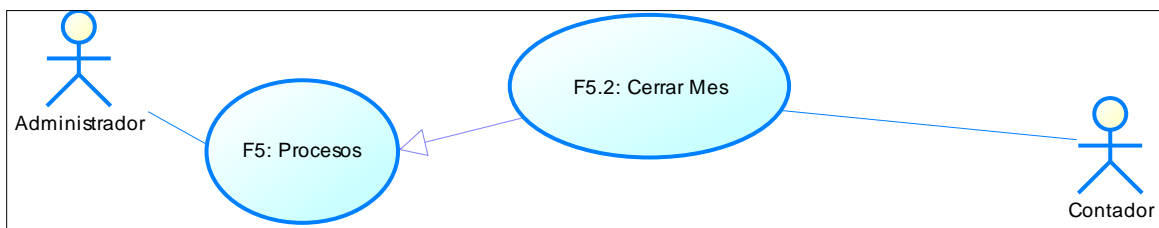


FIG Anexos-13 Diagrama de Casos de Uso – Cerrar Mes [A]

F5.3: Cierre Anual

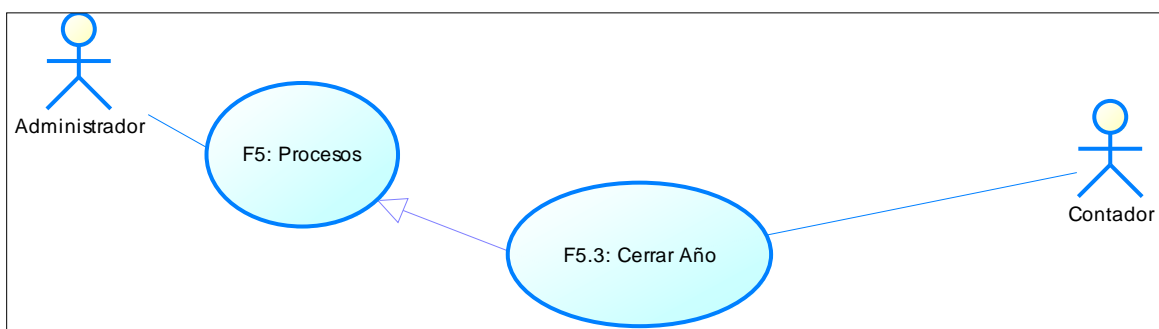


FIG Anexos-13 Diagrama de Casos de Uso – Cerrar Año [A]

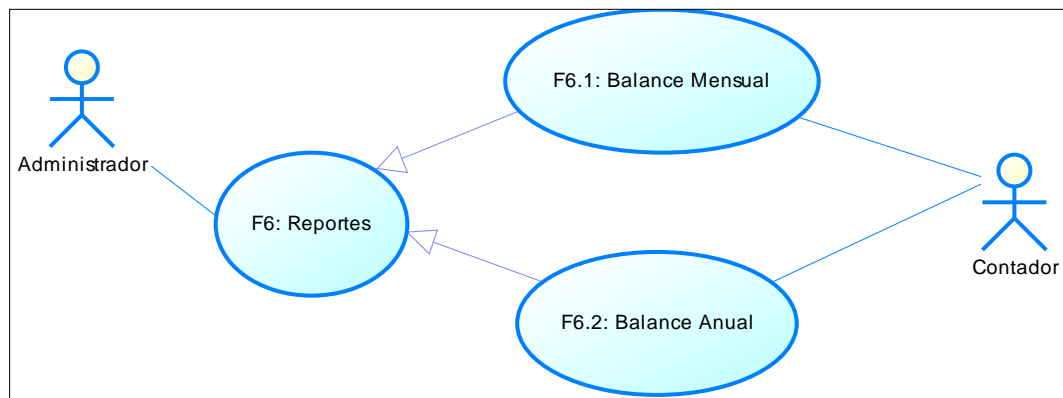
F6: Reportes

FIG Anexos-14 Diagrama de Casos de Uso – Reportes [A]

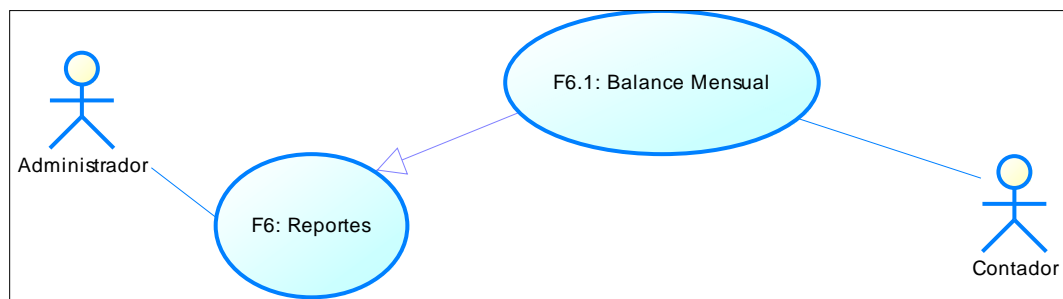
F6.1: Balance Mensual

FIG Anexos-15 Diagrama de Casos de Uso – Balance Mensual [A]

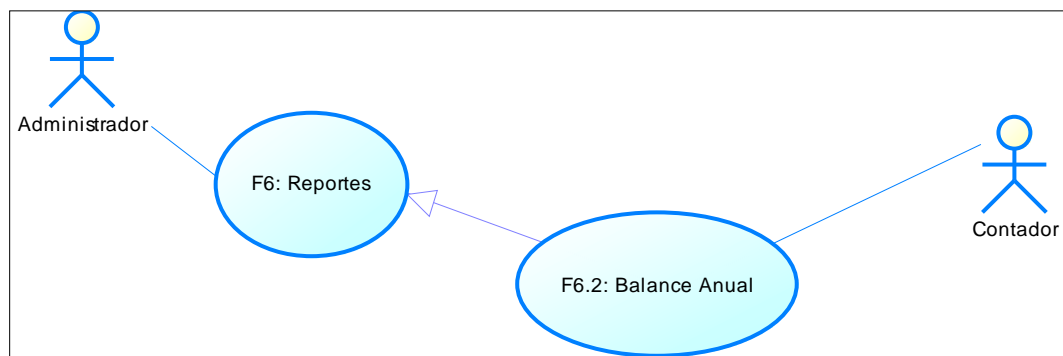
F6.2: Balance Anual

FIG Anexos-16 Diagrama de Casos de Uso – Balance Anual [A]